

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

институт

Кафедра информатики

кафедра

УТВЕРЖДАЮ  
Заведующий кафедрой

А. С. Кузнецов

подпись

инициалы, фамилия

« \_\_\_\_\_ » \_\_\_\_\_ 2019 г.

**БАКАЛАВРСКАЯ РАБОТА**

09.03.04 Программная инженерия

код - наименование направления

Автоматизация учета и обработки клиентских заказов в компании «3ДКоре»

тема

Руководитель

доцент, канд. техн. наук

В. В. Тынченко

подпись, дата

должность, ученая степень

инициалы, фамилия

Выпускник

Н. В. Бабанин

подпись, дата

инициалы, фамилия

Нормоконтролер

доцент, канд. техн. наук

О. А. Антамошкин

подпись, дата

должность, ученая степень

инициалы, фамилия

Красноярск 2019

## **РЕФЕРАТ**

Выпускная квалификационная работа по теме «Автоматизация учета и обработки клиентских заказов в компании «3Д Коре» содержит 58 страниц текстового материала, 25 рисунков, 15 таблиц, 15 использованных источников.

Цель выпускной квалификационной работы заключается в автоматизации бизнес-процессов обработки и учёта клиентских заказов в компании предприятия «3Д Коре» посредством разработки и последующего внедрения веб-приложения.

Выпускная квалификационная работа содержит введение, 2 главы, заключение, список использованных источников.

Первая глава содержит общие сведения об анализе предметной области, обосновании внедрения программного средства, а также анализ похожих программных средств.

Вторая глава включает проектирование программного средства, описание архитектуры и используемых модулей при разработке.

Третья глава содержит описание функциональности и работы приложения.

Заключение содержит анализ полученных результатов.

## Содержание

Введение .....	5
1 Анализ предметной области.....	7
1.1 Обоснование необходимости внедрения программного средства 9	
1.2 Требования к программному средству .....	10
1.3 Анализ существующего программного обеспечения .....	11
1.4 Выбор операционной системы .....	14
1.5 Выбор СУБД .....	17
1.6 Выбор языка программирования.....	19
1.7 Выбор интегрированной среды разработки .....	24
2 Разработка автоматизированной системы .....	27
2.1 Концептуальное проектирование базы данных .....	27
2.2 Логическое проектирование базы данных .....	29
2.3 Физическое проектирование базы данных.....	32
2.4 Проектирование программного обеспечения .....	34
2.4.1 Структура программной системы .....	40
2.5 Руководство программиста.....	42
2.5.1 Требования к техническому обеспечению .....	43
2.5.2 Требования к программному обеспечению .....	43
2.5.3 Установка системы .....	44
2.5.4 Общие характеристики веб-приложения .....	45
2.5.5 Работа с веб-приложением на стороне сервера .....	46
2.5.6 Внесение изменений в веб-приложение.....	46

2.6	Руководство пользователя.....	47
2.7	Тестирование программного средства.....	51
	Заключение.....	55
	Список использованных источников.....	57

## **Введение**

На сегодняшний день роль корпоративных веб-приложений в сфере предоставления услуг занимает одно из ведущих мест. В 2019 году невозможно представить онлайн торговлю без корпоративного портала или административной панели сайта, в которой руководитель может, например, построить отчёт о прибыли или оказанных услугах за выбранный период. Современные технологии, позволяют расширить функционал не только до построения разнообразных отчётов или шаблонов документов. Улучшениями могут быть автоматизация постановки задач для исполнителя, контроль процесса со стороны руководителя, возможность управлять контентом на основе статистических показателей. Импортировать и экспортировать контент в социальные сети, обмениваться с клиентами посредством встраиваемых мессенджеров. Возможности можно перечислять бесконечно. Но главное, что их объединяет, способность сократить время трудозатрат, которое чаще расходуется на рутинные задачи.

В данной работе будет рассмотрен процесс автоматизации учёта и обработки заказов в компании, которая специализируется на печати трёхмерных изделий.

Цель выпускной квалификационной работы заключается в автоматизации бизнес-процессов учёта и обработки заказов, поступающих как из интернет-магазина, так и при поступлении заказов на почтовый ящик сотрудников компании или по телефону. Данная цель будет достигнута посредством разработки и последующего внедрения веб-приложения.

Для достижения поставленной цели требуется решить следующие задачи:

- провести анализ предметной области и обосновать необходимость внедрения программного средства для оптимизации работы службы печати трёхмерных изделий;
- сформулировать постановку задачи автоматизации;

- исследовать рынок существующих программных продуктов аналогичного назначения;
- обосновать выбор средств реализации проекта;
- произвести проектирование и реализацию программного средства.

## 1 Анализ предметной области

Основным направлением деятельности компании «3Д Коре» является проектирование и разработка 3Д принтеров. В структуру компании включены два подразделения: «3Д Китчен», основной задачей которого является производство и продажа форм для пряников и печения, и «3Д Центр», который предоставляет услуги по печати трёхмерных изделий на заказ. Компания расположена в 2 офисах в городе Красноярск. Штат сотрудников компании, насчитывает около 10 человек.

На сегодняшний день компания не располагает средствами автоматизации бизнес-процессов. Выполнение шагов для решения повседневных задач происходит вручную, либо в виде письменных распоряжений, либо в виде устных.

На первоначальном этапе анализа был построен ряд диаграмм. Организационная диаграмма, представленная на рисунке 1, отражает ролевую структуру управления компанией.

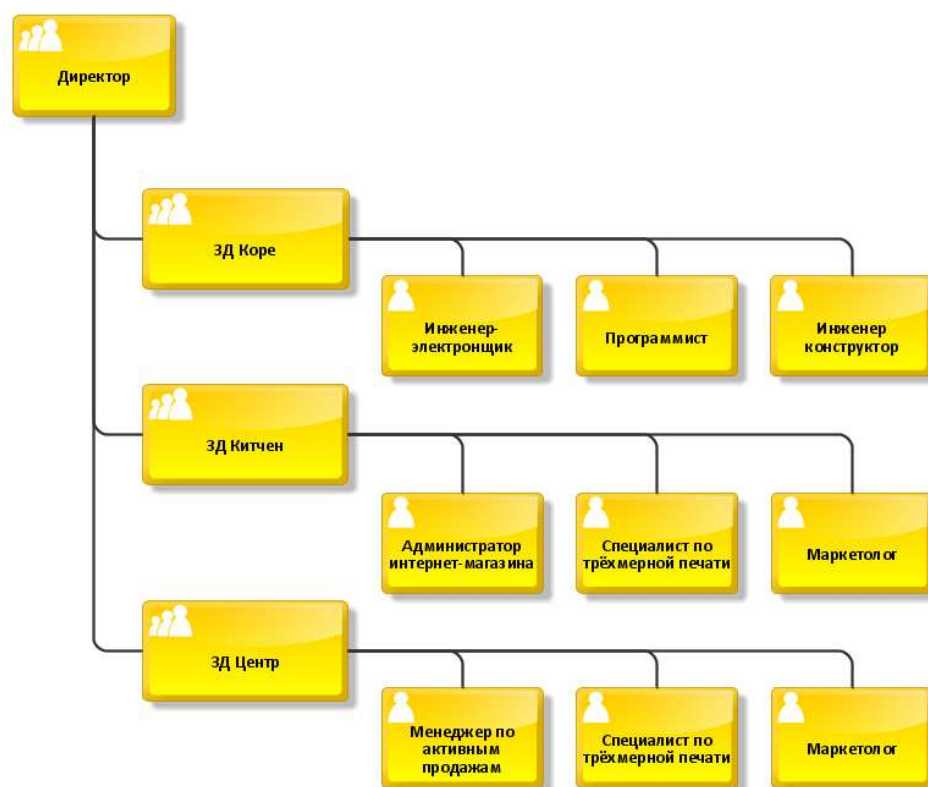


Рисунок 1 – Организационная диаграмма

VAD-диаграмма, представленная на рисунке 2, показывает основные звенья бизнес-процесса верхнего уровня



Рисунок 2 – VAD – диаграмма

eEPC-диаграмма (рисунок 3, 4) детально отражает процесс компании при обработке клиентских заказов, что заказ поступает в компанию от клиента непосредственно директору. В свою очередь директор формирует распоряжение в письменном виде на исполнителя (специалиста по трёхмерной печати), а далее уже исполнитель составляет отчёт о выполнении и завершении работы над заказом. Корректировки и уведомления участников процесса также происходят либо форме электронного письма, либо устно.

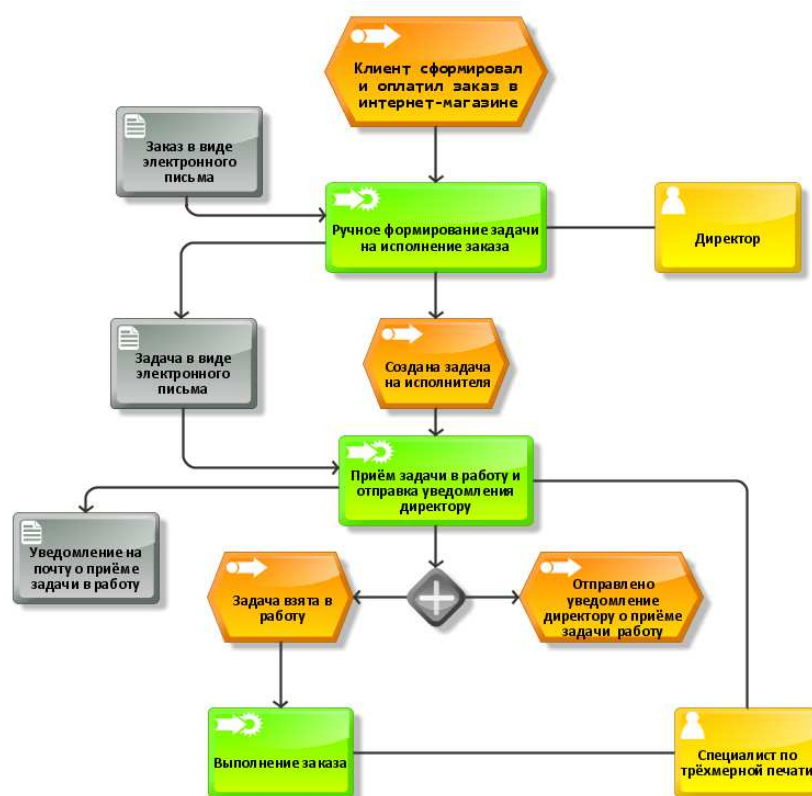


Рисунок 3 – eEPC диаграмма приёма и работы над заказом



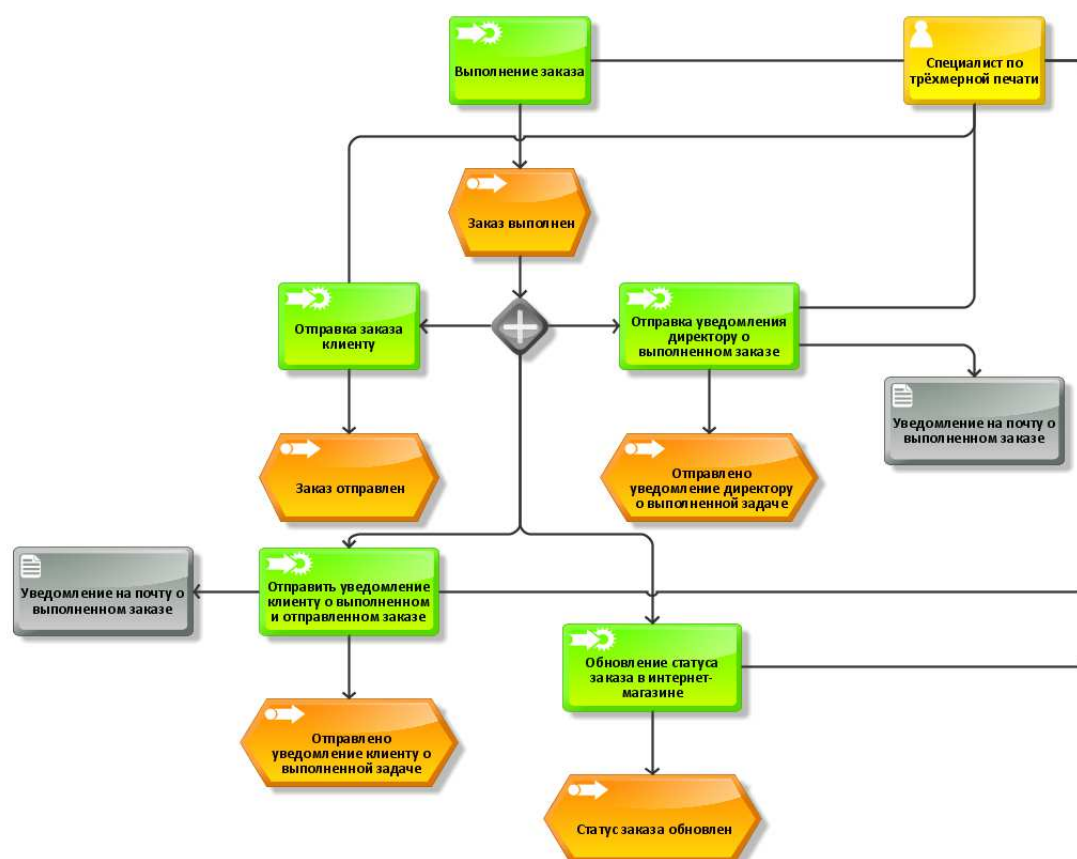


Рисунок 4 – eEPC диаграмма выполнения заказа и отправки его клиенту

### 1.1 Обоснование необходимости внедрения программного средства

На основании анализа предметной области можно сделать вывод, что при увеличении количества клиентских заказов трудозатраты на их реализацию могут значительно увеличиться.

Следующие повседневные задачи могут также повысить риск невыполнения обязательств перед клиентом:

- определение последовательности выполнения заказов, в том числе если приоритет меняется;
- при необходимости внести корректировки в заказ руководитель должен дополнительно уведомить специалиста о корректировке заказа;

- при необходимости связаться с администратором для решения возникших вопросов;
- при необходимости передать данные о заказе руководителю для решения организационных или иных возникших вопросов;
- не забыть отчитаться о сделанной работе для обеспечения корректной отчётности.

Решением всех этих проблем руководителю и специалисту, зачастую, приходится разбираться в процессе изготовления изделий и одновременной коммуникации с клиентами, что влечет за собой увеличение количество ошибок и риск снизить качество работы. Разрабатываемое приложение призвано сделать процесс более прозрачным и последовательным. Корректировка заказа не должна вызывать путаницы и значительно увеличивать трудозатраты сотрудников. За счет этого будет повышена производительность сотрудников благодаря экономии времени, а также снижению трудоемкости операций.

## **1.2 Требования к программному средству**

С учетом всех факторов к программному средству были предъявлены следующие требования:

- отображение общего списка задач в сокращенном виде;
- отображение полной информации задаче с заказом в отдельном окне;
- возможность обеспечить интеграцию с существующим интернет-магазином компании;
- ведение учета выполненных заказов, сбор статистики;
- уведомление исполнителя о поступившем заказе;
- авторизация в приложении;
- хранение данных о пользователях;

- отображение списка пользователей;
- у каждого пользователя должен быть личный кабинет;
- оформление внешнего вида приложения в соответствии со стилистикой компании;
- наличие интуитивно-понятного удобного для пользователя интерфейса;
- обработка исключительных ситуаций, отказоустойчивость программного средства.

### 1.3 Анализ существующего программного обеспечения

В процессе поиска схожих программных продуктов были выделены два решения от разных производителей, а именно:

- «amoCRM» от компании «QSOFТ»;
- «bpm'online»;
- «Flowlu».

«amoCRM»

На рисунке 5 приведён скриншот страницы, на которой располагаются списки задач, оформленные в виде колонок.

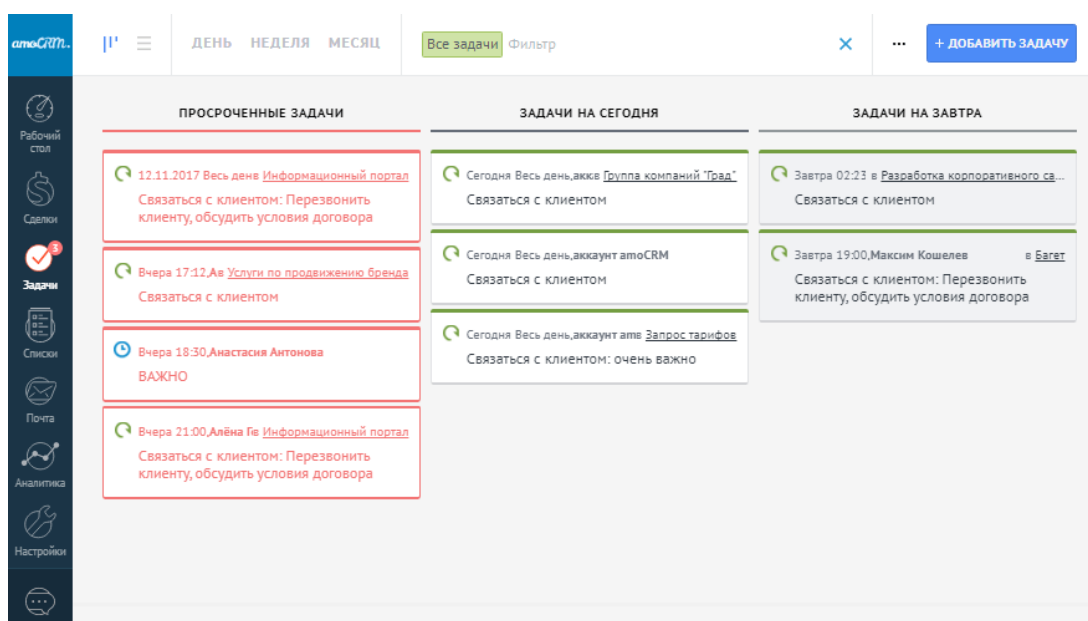


Рисунок 5 – Скриншот страницы с списками задач в разных статусах приложения «amoCRM»

Программное средство «atoCRM» обладает достаточно богатым разнообразием решаемых задач:

- авторизация и аутентификация пользователей;
- разделение ролей;
- ведение задач;
- возможность интегрироваться с внешними системами;
- ведение базы клиентов и контрагентов;
- ведение событий;
- создание пользовательских виджетов.

### «bpm'online»

Скриншот страницы с детальной информацией о задаче (рисунок 6) отражает набор разнообразных полей для ввода информации.

The screenshot displays the 'bpm'online' web interface for a task titled 'Согласовать пакет документов' (Agree on a package of documents). The interface includes a header with a search bar, a 'bpmonline' logo, and a 'вид' (view) dropdown. Below the header, there is a task summary section with fields for 'Заголовок' (Title), 'Начало' (Start), 'Завершение' (End), 'Состояние' (Status), 'Ответственный' (Responsible), 'Автор' (Author), 'Приоритет' (Priority), and 'Категория' (Category). The task is currently in 'В работе' (In progress) status, with a start date of 02.04.2018 and an end date of 05.04.2018. The responsible person is 'Евгений Мирный' (Eugene Mironov) and the category is 'Встреча' (Meeting). Below the summary, there is a tabbed interface with tabs for 'ОСНОВНАЯ ИНФОРМАЦИЯ' (Main Information), 'УЧАСТНИКИ' (Participants), 'ФАЙЛЫ И ПРИМЕЧАНИЯ' (Files and Comments), 'EMAIL', 'ЗВОНКИ' (Calls), and 'ЛЕНТА' (Feed). The 'ОСНОВНАЯ ИНФОРМАЦИЯ' tab is active, showing a 'Результат' (Result) section with a 'Результат' (Result) field and a 'Результат подробно' (Result in detail) section. Below this is a 'Связи' (Links) section with various fields for 'Договор' (Contract), 'Заказ' (Order), 'Контакт' (Contact), 'Конфигурационная единица' (Configuration unit), 'Мероприятие' (Event), 'Проблема' (Problem), 'Релиз' (Release), 'Проект/работа' (Project/work), 'Документ' (Document), 'Изменение' (Change), 'Контрагент' (Counterparty), 'Лид' (Lead), 'Обращение' (Inquiry), 'Продукт' (Product), and 'Счет' (Account). At the bottom, there is a 'Напоминания' (Reminders) section with fields for 'Ответственному' (Responsible), 'Автору' (Author), 'Дата напоминания' (Reminder date), and 'Время напоминания' (Reminder time).

Рисунок 6 – Скриншот страницы с задачей приложения «bpm'online»

Основные возможности:

- Построение бизнес-процессов;
- Настройка визуального интерфейса;
- Работа с задачами;
- Работа с почтой.

Недостатки:

- Высокая цена на подписку с необходимой функциональностью;
- Нет возможности связать задачу с заказом.

«Flowlu»

На рисунке 7 приведён пример уведомлений, который показывает какие действия были выполнены над задачей за определенный период времени

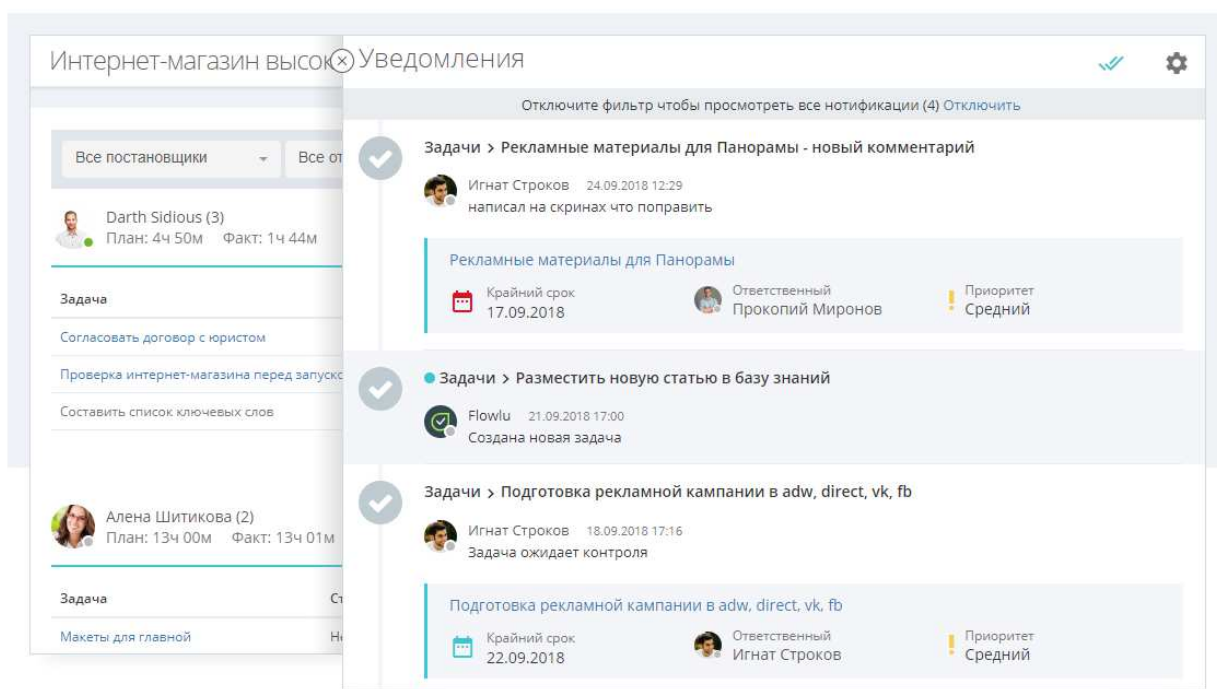


Рисунок 7 – Скриншот страницы приложения «Flowlu»

Основные возможности:

- Совместная работа над задачами;
- Регулярные задачи и шаблоны;
- Учёт времени по задачам;

- Аналитика и отчёты по задачам.

Недостатки:

- Нет возможности гибко интегрироваться с интернет-магазином;
- Высокая цена по подписке;
- Нет возможности развернуть на собственном сервере.

Проведя анализ данных программных решений, можно сделать вывод, что представленные CRM системы и системы управления задачами обладают широким набором разнообразных функций. С одной стороны, это огромный плюс, потому что такие системы требуют минимального количества доработок в начале. Однако, чтобы внести изменения в бизнес-логику придётся детально разобраться не только в концепции готового решения, но и освоить один или несколько языков программирования. Внесение дополнений в существующее решение, перегруженное функциональностью, которое не используется и наполовину, может оказаться убыточным для компании.

Таким образом, можно сделать вывод, что целесообразно разработать собственную систему автоматизации процесса обработки заказов, которая в первую очередь будет решать две ключевые задачи: интегрироваться с интернет-магазином и предоставлять функции, связанные с клиентскими заказами.

#### **1.4 Выбор операционной системы**

Проанализировав наиболее популярные операционные системы для возможности размещения на них веб-сервера с системой обработки клиентских заказов, для дальнейшего исследования были выбраны три:

- Windows Server;
- Solaris;
- Linux.

Исследования, которые проводила компания Тэглайн с 2014 по 2016 (рисунок 8) показали, что наиболее популярными для сервера, являются операционные системы семейства Linux, такие как: Debian, Ubuntu Server, CentOS, FreeBSD.

Рейтинг серверных операционных систем 2016

#	Название	Год	Дистрибутив Linux	Платная лицензия	Платная лицензия	Open source
1 +1	 <a href="#">Debian</a>	1993	✓	✓	×	✓
1 +3	 <a href="#">Ubuntu Server</a>	2004	✓	✓	×	✓
2 -1	 <a href="#">CentOS</a>	2004	✓	✓	×	✓
3 -1	 <a href="#">FreeBSD</a>	1993	×	✓	×	✓
3 -2	 <a href="#">Windows Server</a>	1993	×	×	от \$501	×
4 +2	 <a href="#">Red Hat Enterprise Linux</a>	2000	✓	✓	от \$349 / год	✓
5 0	 <a href="#">Fedora</a>	2003	✓	✓	×	✓
5 +2	 <a href="#">Gentoo</a>	2002	✓	✓	×	✓
5 +1	 <a href="#">OS X Server</a>	1999	×	×	1.499 р.	×
6 new	 <a href="#">OpenBSD</a>	1995	×	✓	×	✓
6 new	 <a href="#">Oracle Linux</a>	2006	✓	✓	×	✓

Рисунок 8 – Рейтинг серверных операционных систем

Немаловажным аспектом при выборе операционной системы является и цена. Серверная версия операционной системы Windows является платной, в то время как Linux, а именно дистрибутивы Debian, Ubuntu Server, CentOS предоставляется под бесплатной лицензией.

Еще одним важным пунктом, набирающим популярность в России, является импортозамещение. Операционная система Windows была разработана и поддерживается компанией США и является закрытым программным продуктом. В то же время операционные системы Debian, Ubuntu Server, CentOS, основанные на ядре Linux, разрабатываются сообществами разработчиков из разных стран, в том числе из России, а сами операционные системы имеют открытые исходные коды.

Также в поддержку импортозамещения следует добавить, что в России также ведутся разработки отечественных операционных систем, в основном на ядре Linux. Некоторые представлены в списке:

- «Альт Линукс СПТ» от компании ООО «Свободные программы и технологии»;
- «ОСъ» от Национального центра информатизации (входит в госкорпорацию «Ростех»);
- «Astra Linux» от НПО «Русские базовые информационные технологии» (РусБИТех);
- «ROSA Linux» от ООО «НТЦ ИТ РОСА»;
- «Альфа ОС» от компании ALFA Vision.

Таким образом, преимуществом выбора ОС Linux будет возможность быстрой миграции веб-сервера приложения между операционными системами семейства Linux

Проведя анализ, можно сделать вывод, что наиболее предпочтительным выбором операционной системы для продуктивной среды



заказчика, на которую будет устанавливаться разрабатываемое программное обеспечение, будет ОС CentOS на базе ядра Linux.

## 1.5 Выбор СУБД

Несмотря на то, что все системы управления базами данных выполняют одну и ту же основную задачу (т.е. дают возможность пользователям создавать, редактировать и получать доступ к информации, хранящейся в базах данных), сам процесс выполнения этой задачи варьируется в широких пределах. Кроме того, функции и возможности каждой СУБД могут существенно отличаться. Различные СУБД документированы по-разному: более или менее тщательно. По-разному предоставляется и техническая поддержка.

При сравнении различных популярных систем, следует учитывать, удобна ли для пользователя и масштабируема ли данная конкретная СУБД, а также убедиться, что она будет хорошо интегрироваться с другими продуктами, которые уже используются. Кроме того, во время выбора следует принять во внимание стоимость системы и поддержки, предоставляемой разработчиком [6].

Существует несколько популярных СУБД, как платных, так и бесплатных, которые можно рекомендовать для применения в организации.

Для сравнения возьмём наиболее популярные СУБД:

- MySQL;
- Microsoft SQL Server;
- PostgreSQL.

### MySQL

Достоинства:

- Распространяется бесплатно;
- Прекрасно документирована;
- Предлагает много функций, даже в бесплатной версии;

- Пакет MySQL включен в стандартные репозитории наиболее распространённых дистрибутивов операционной системы Linux, что позволяет устанавливать её элементарно;

- Поддерживает набор пользовательских интерфейсов;
- Может работать с другими базами данных, включая DB2 и Oracle.

Недостатки:

- Придётся потратить много времени и усилий, чтобы заставить MySQL выполнять несложные задачи, хотя другие системы делают это автоматически, например: создавать инкрементные резервные копии;

- Отсутствует встроенная поддержка XML или OLAP;
- Для бесплатной версии доступна только платная поддержка.

Идеально подходит для: организаций, которым требуется надежный инструмент управления базами данных, но бесплатный.

### Microsoft SQL сервер

Достоинства:

- Продукт очень прост в использовании
- Текущая версия работает быстро и стабильно
- Движок предоставляет возможность регулировать и отслеживать уровни производительности, которые помогают снизить использование ресурсов.

- Вы сможете получить доступ к визуализации на мобильных устройствах;

- Он очень хорошо взаимодействует с другими продуктами Microsoft.

Недостатки:

- Цена для юридических лиц оказывается неприемлемой для большей части организаций;

- Даже при тщательной настройке производительности корпорация SQL Server способен занять все доступные ресурсы;

- Сообщается о проблемах с использованием службы интеграции для импорта файлов [6].

Идеально подходит для: крупных организаций, которые уже используют ряд продуктов Microsoft.

## PostgreSQL

Достоинства:

- Является масштабируемым и способен обрабатывать терабайты данных;
- Поддерживает формат json;
- Существует множество predefined функций;
- Доступен ряд интерфейсов.

Недостатки:

- Документация туманна, поэтому, возможно, ответы на некоторые вопросы придется искать в интернете;
- Конфигурация может смутить неподготовленного пользователя;
- Скорость работы может падать во время проведения пакетных операций или выполнения запросов чтения.

Идеально подходит для организаций с ограниченным бюджетом, но квалифицированными специалистами, когда требуется возможность выбрать свой интерфейс и использовать json.

Подведя итог, можно сделать вывод, что оптимальной СУБД для хранения данных в нашем случае является PostgreSQL.

## 1.6 Выбор языка программирования

Несмотря на то, выбор языка программирования Java predefined заказчиком системы, будет оправданным провести сравнительный анализ его с другими языками, такими как Python и C# так как согласно рейтингу, по

версии издания IEEE Spectrum за 2018[1], они являются наиболее популярными при разработке веб-приложений.

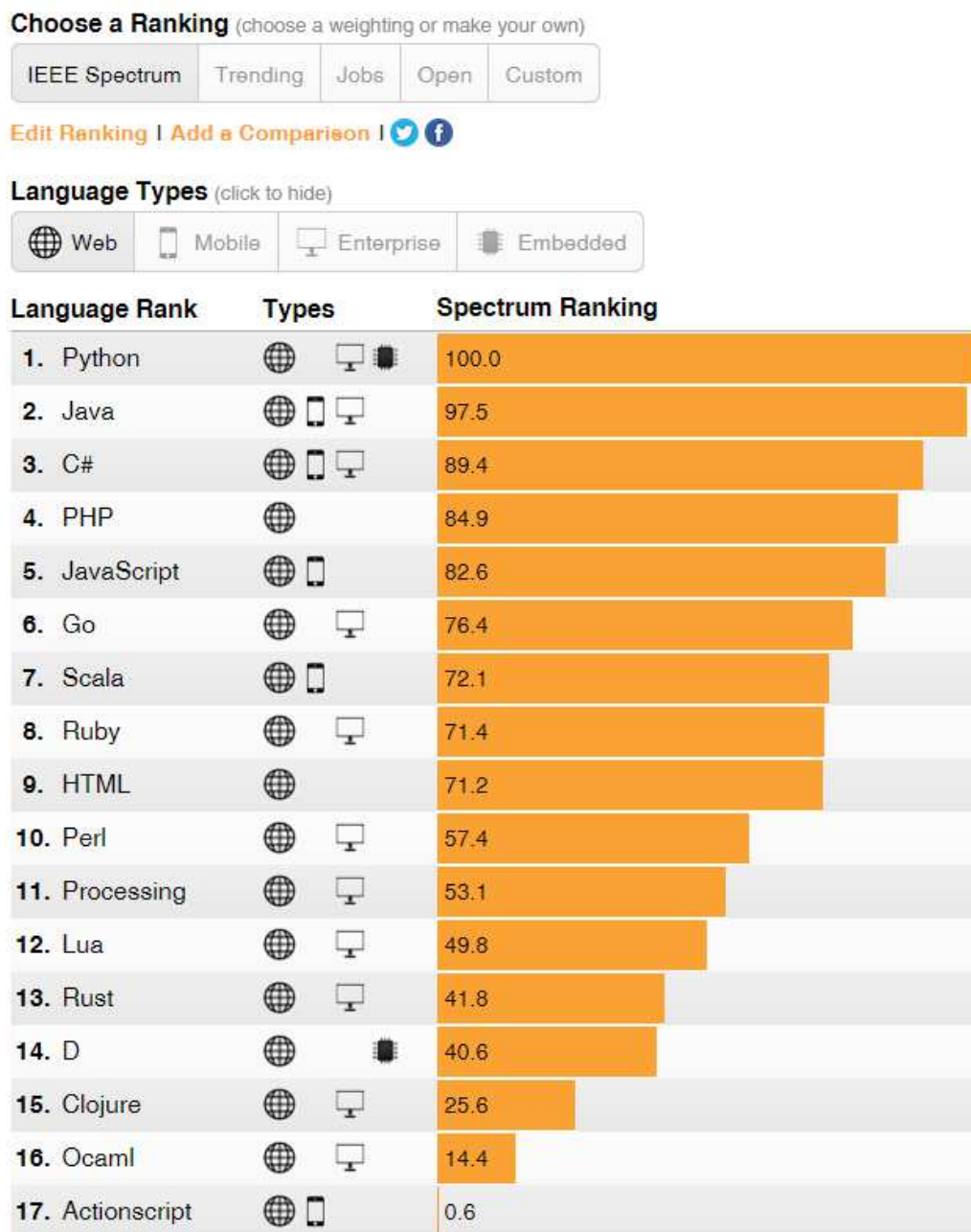


Рисунок 9 – Рейтинг по версии издания IEEE Spectrum

Основные преимущества Java:

- Наиболее популярный по версии ТЮВЕ [2];

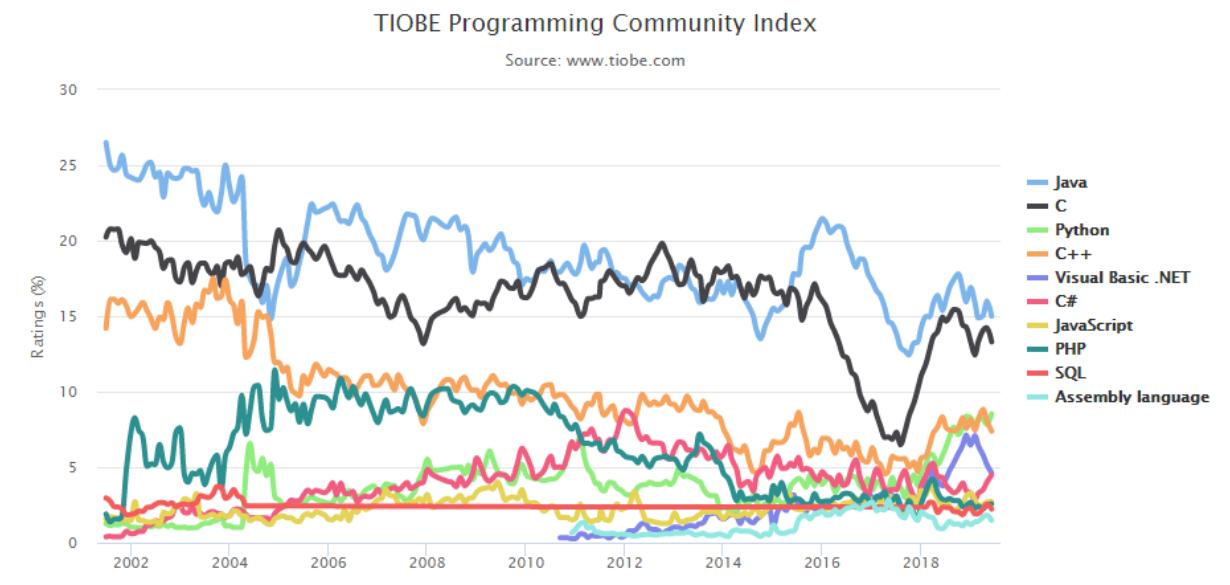


Рисунок 10 – Рейтинг TIOBE Programming Index

- Кроссплатформенный;
- Строго типизированный;
- Исходники среды выполнения открыты, и поддерживаются большим количеством разработчиков по всему миру;
- Широкий выбор фреймворков и библиотек для построения корпоративных приложений;
- Наиболее популярные решения для создания веб-приложений имеют открытый исходный код и распространяются под свободной лицензией. К таким относятся например: Spring Framework, Apache Karaf, GWT.

Недостатки:

- Низкое, в сравнении с другими языками, быстродействие, повышенные требования к объему оперативной памяти (ОП);
- Большой объем стандартных библиотек и технологий создает сложности в изучении языка;
- Постоянное развитие языка вызывает наличие как устаревших, так и новых средств, имеющих одно и то же функциональное назначение.

#### Основные преимущества C#:

- поддержка Microsoft. В отличие от Java, которой не пошел на пользу переход в собственность Oracle, C# хорошо развивается благодаря усилиям Microsoft;
- в последнее время много совершенствуется. Так как C# был создан позже, чем Java и другие языки, то требовалось очень много доработать;
- много синтаксического сахара. Синтаксический сахар – конструкции языка программирования, которые созданы для облегчения написания и понимания кода и не играют роли при компиляции;
- средний порог вхождения. Синтаксис похожий на C, C++ или Java облегчает переход для других программистов;
- Наличие дополнительного функционального языка программирования (F#).

#### Недостатки:

- ориентированность, в основном, только на .NET (на Windows платформу);
- Бесплатная лицензия только для небольших компании, учащихся и программистов, работающих над собственными проектами в одиночку. Для больших проектов команд покупка лицензий потребует значительных материальных вложений.

#### Основные преимущества Python:

- Обширные библиотеки поддержки;
- Наличие встроенных инструментов для интеграции. Python интегрирует Enterprise Application Integration, которая облегчает разработку веб-сервисов, вызывая компоненты COM или COBRA. Он имеет мощные возможности управления, так как он вызывает напрямую через C, C++ или Java через Jython;

- Улучшенная производительность программиста. Благодаря мощным функциям интеграции процессов, модульному тестированию и расширенным возможностям управления способствуют увеличению скорости для большинства приложений и производительности приложений. Это отличный вариант для построения масштабируемых многопротокольных сетевых приложений.

Недостатки:

- Трудность в использовании других языков. Любители Python настолько привыкли к его функциям и обширным библиотекам, что сталкиваются с проблемами при изучении или работе на других языках программирования. Эксперты по Python могут рассматривать объявление «значений» или переменных «типов», синтаксических требований добавления фигурных скобок или точек с запятой как обременительную задачу;

- Python выполняется с помощью интерпретатора вместо компилятора, что приводит к его замедлению, потому что компиляция и выполнение помогают ему работать нормально. С другой стороны, видно, что это быстро для многих веб-приложений;

- По сравнению с популярными технологиями, такими как JDBC и ODBC, уровень доступа к базе данных в Python оказался немного недоразвитым и примитивным.

Таким образом, проведя анализ языков программирования, можно считать, что первоначальный выбор Java был корректным, в связи с тем, что данный язык обладает следующими важными преимуществами:

- Значительной поддержкой со стороны разработчиков, не привязанных к какой-то конкретной корпорации, как например C# привязан к Microsoft;

- Исходные коды среды выполнения Java открыты, что позволяет не только детально разобраться с архитектурой языка, но и вносить доработки для интеграции с новыми возможностями языка;
- Для разработки веб-приложений очень важна тесная интеграция с системами хранения, не только файловыми, но и СУБД. Здесь Java также имеет значительное преимущество так как для всех популярных реляционных баз данных имеет библиотеки JDBC (Java Data Base Connectivity) в отличие от языка Python;
- Способность работать на всех популярных платформах под управлением таких операционных систем как Windows, Linux, MacOS, Solaris и другие. Это немаловажный аспект, так как продуктивная среда может быть развёрнута, на ОС Linux, в то время как разработчики для себя могут выбрать любую комфортную ОС с графической оболочкой, это и Windows 10, и Linux Ubuntu, Linux Kubunu, Linux Mint, MacOS и многие другие;
- Язык Java может быть использован для написания программных продуктов в рамках программы импортозамещения при разработке отечественного ПО.

### **1.7 Выбор интегрированной среды разработки**

Среди наиболее популярных сред разработки на языке Java можно выделить три:

- IntelliJ IDEA;
- Eclipse;
- NetBeans.

IntelliJ IDEA

Преимущества:

- Помимо платной версии (Ultimate edition) с расширенными возможностями, имеет бесплатную версию (Community edition);



- Поддерживает не только последние версии Java, но и Kotlin, Groovy и Scala;
- Работает с такими системами контроля версий как Git, SVN, Mercurial и CVS;
- Есть поддержка популярных фреймворков Java EE, Spring, GWT, Vaadin, Play, Grails;
- Есть встроенный инструментарий для подключения к СУБД используя драйверы JDBC. Также есть возможность выполнять SQL-запросы, просматривать/добавлять/менять/удалять данные в таблицах;
- Помимо подсветки синтаксиса и автоподстановки имеет богатый инструментарий по рефакторингу кода, анализу кода;
- Тесно интегрируется с такими системами сборки как Ant, Maven и Gradle, позволяет запускать программу прямо из исходного кода, не обращаясь к терминалу;
- Наличие распознавания контекста в исходном коде. Например, при написании в файле с расширением java, внедрён фрагмент SQL запроса, IDE распознает и позволит не только в нем подсветить ключевые слова и выполнить автоподстановку при редактировании, но и позволит запустить его прямо оттуда, где он находится, используя контекстное меню [8].

Основным недостатком данной среды разработки можно считать цену, которую придётся заплатить за расширенный функционал Ultimate edition.

## Eclipse IDE

### Преимущества:

- IDE построена на базе спецификации динамической модульной системы. Иными словами, среда разработки представляет из себя модульную платформу, к которой присоединяются плагины для разработки на различных языках, в том числе Java, а также различные инструменты интеграции;
- Eclipse IDE полностью бесплатна;

- Возможность установить самые разнообразные плагины: для интеграции с системами контроля версий, системами сборки проекта, построение отчётов, инструменты для моделирования бизнес-процессов и т.д.;

- Так как среда разработки написана на Java, является кроссплатформенной;

- Возможность настраивать отображение панелей в зависимости от контекста работы.

Недостатки:

- Большое количество плагинов значительно замедляют работу среды

- Так как плагины зачастую пишутся сторонними разработчиками, устанавливая дополнительные модули, нередко можно получить их конфликт между собой;

- Плагины зачастую нестабильны или обладают небольшим количеством функциональности.

NetBeans

Преимущества:

- «Из коробки» включает в себя инструментарий для работы с Java, JavaScript и HTML. А также поддержку интеграции с такими веб-серверами как GlassFish и Tomcat;

- Редактор позволяет отслеживать ошибки «на лету» при вводе текста;

- Скорость работы не зависит от количества установленных плагинов. Плагины не конфликтуют друг с другом;

- В инструментарий NetBeans входят графические редакторы для построения пользовательского интерфейса.

Из недостатков можно отметить бедный набор инструментов в отношении интеграции с СУБД, Системами контроля версий, системами сборки, в сравнении с IntelliJ IDEA и в Eclipse IDE.

Как итог, можно сказать, что наиболее целесообразно использовать IntelliJ IDEA Community Edition в связи с наличием у неё достаточно разнообразного инструментария для работы с системами управления версиями и системами сборки, возможности работать с базой данных напрямую, не устанавливая дополнительного ПО. Одним из аргументов также следует отметить и производительность. В отличие от Eclipse IDE, система IntelliJ IDEA уже включает необходимый инструментарий, который не конфликтует между собой и не замедляет работу приложения.

## **2 Разработка автоматизированной системы**

### **2.1 Концептуальное проектирование базы данных**

В соответствии с предметной областью система строится с учётом следующих особенностей:

- Каждая задача создаётся в рамках одного заказа;
- Заказ может принадлежать только одной задаче;
- У каждой задачи может быть только один исполнитель;
- Каждый заказ может иметь несколько товаров;
- Каждый товар может быть связан только с одним заказом;
- Каждый пользователь может иметь несколько ролей.

Выделим базовые сущности этой предметной области:

- Сотрудник компании. Атрибуты сотрудников: ФИО, электронная почта, номер телефона;
- Роль сотрудника. Атрибуты роли: наименование роли;

- Задача. Атрибуты задачи: заголовок, описание задачи, сотрудник-исполнитель, связанный заказ;
- Заказ. Атрибуты заказа: ФИО, электронная почта, номер телефона клиента, способ доставки, адрес доставки, служба доставки;
- Товар. Атрибуты товара: артикул, наименование, наименование категории, цвет, размер.

На рисунке 11 отражена ER-диаграмма логической структуры данных приложения.

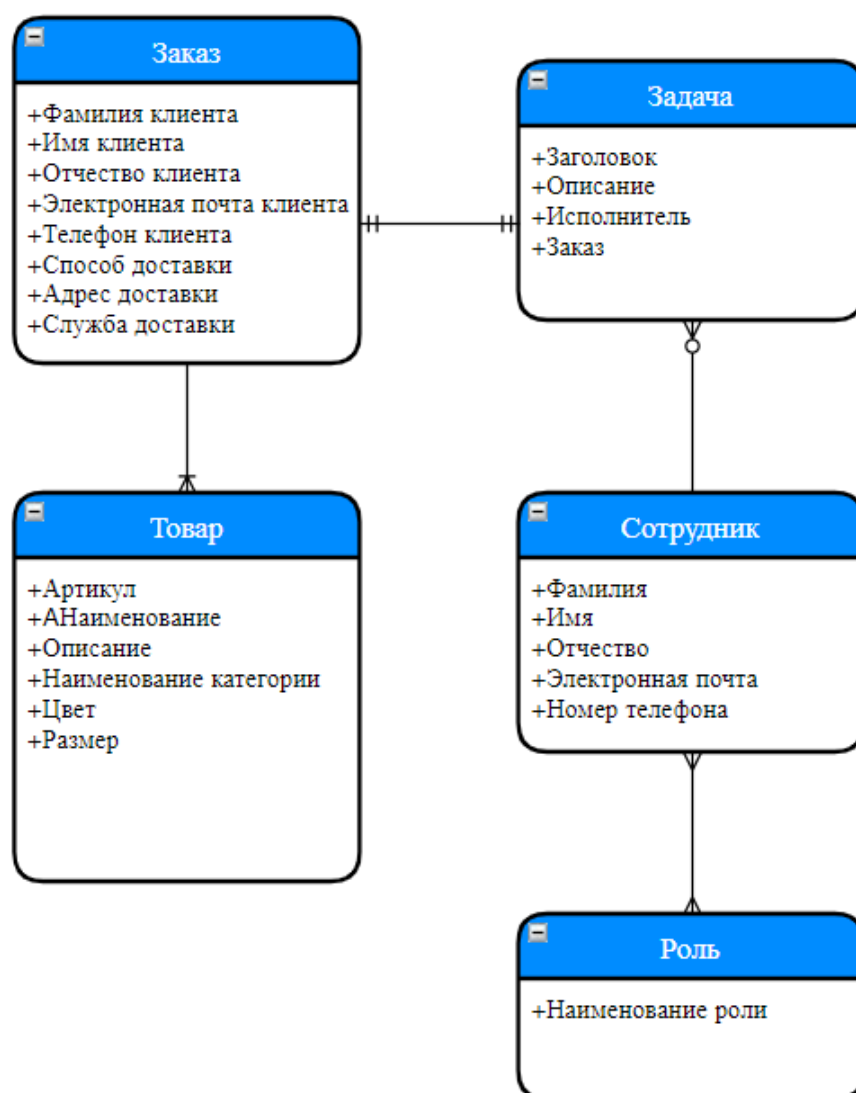


Рисунок 11 – ER-модель базы данных

## 2.2 Логическое проектирование базы данных

### Преобразование ER-диаграммы в схемы БД

Таблицы 1,2,3,4,5,6 отражают наборы полей в сущностях базы данных, а также отношения их между собой.

Таблица 1 – Схема отношения Сотрудник

Содержание поля	Имя поля	Тип, длина	Примечание
Идентификатор	Id	Big Integer	Первичный ключ
Имя	FirstName	Variable Character (255)	Обязательное поле
Фамилия	SecondName	Variable Character (255)	Обязательное поле
Отчество	Patronymic	Variable Character (255)	
Электронная почта	Email	Variable Character (255)	Обязательное поле
Телефон	Phone	Variable Character (64)	

Таблица 2 – Схема отношения Роль

Содержание поля	Имя поля	Тип, длина	Примечание
Идентификатор	Id	Big Integer	Первичный ключ
Наименование	Name	Variable Character (128)	Обязательное поле

Таблица 3 – Схема отношения Сотрудник-Роль

Содержание поля	Имя поля	Тип, длина	Примечание
Идентификатор	Id	Big Integer	Первичный ключ

Продолжение таблицы 3

Идентификатор сотрудника	UserId	Big Integer	Внешний ключ к «Сотрудник»
Идентификатор роли	RoleId	Big Integer	Внешний ключ к «Роль»

Таблица 4 – Схема отношения Задача

Содержание поля	Имя поля	Тип, длина	Примечание
Идентификатор	Id	Big Integer	Первичный ключ
Заголовок	Title	Variable Character (512)	
Описание	Description	Variable Character (2000)	
Статус	Status	Variable Character (16)	Обязательное поле
Идентификатор исполнителя	AssigneeId	Big Integer	Внешний ключ к «Сотрудник»
Идентификатор заказа	CartId	Big Integer	Внешний ключ к «Заказ»

Таблица 5 – Схема отношения Заказ

Содержание поля	Имя поля	Тип, длина	Примечание
Идентификатор	Id	Big Integer	Первичный ключ
Имя клиента	CustomerFirstName	Variable Character (255)	Обязательное поле

Продолжение таблицы 5

Фамилия клиента	CustomerSecondName	Variable Character (255)	Обязательное поле
Отчество клиента	CustomerPatronymic	Variable Character (255)	
Электронная почта клиента	CustomerEmail	Variable Character (255)	Обязательное поле
Телефон клиента	CustomerPhone	Variable Character (64)	Обязательное поле
Способ доставки	DeliveryMethod	Variable Character (16)	Обязательное поле
Адрес доставки	DeliveryAddress	Variable Character (512)	
Служба доставки	DeliveryService	Variable Character (128)	

Таблица 6 – Схема отношения Товар

Содержание поля	Имя поля	Тип, длина	Примечание
Идентификатор	Id	Big Integer	Первичный ключ
Артикул	Code	Variable Character (64)	Обязательное поле
Наименование	Title	Variable Character (512)	
Описание	Description	Variable Character (2000)	
Наименование категории	CategoryName	Variable Character (64)	
Цвет	Colour	Integer	Обязательное поле

## Продолжение таблицы 6

Размер	Size	Variable Character (8)	Обязательное поле
Идентификатор заказа	CartId	Big Integer	Внешний ключ к «Заказ»

Составление схем реляционных отношений: каждое реляционное отношение соответствует одной сущности и в него вносятся все атрибуты этой сущности. Для каждого отношения определяются первичный ключ и внешние ключи (в соответствии со схемой БД). В том случае, если базовое отношение не имеет потенциальных ключей, вводится суррогатный первичный ключ, который не несёт смысловой нагрузки и служит только для идентификации записей.

### 2.3 Физическое проектирование базы данных

При использовании СУБД PostgreSQL последовательность создания объектов БД, следующая:

1. Создание пользователей (create user);
2. Создание БД (create database);
3. Создание кластеров и таблиц (create table);
4. Создание последовательностей (create sequence).

Следующие шаги можно выразить на языке SQL

Создание БД (create database).

```
CREATE DATABASE 3dcore_db
WITH
OWNER = admin_db
ENCODING = 'UTF8'
LC_COLLATE = 'English_United States.1252'
```



```
LC_CTYPE = 'English_United States.1252'  
TABLESPACE = pg_default  
CONNECTION LIMIT = -1;
```

Создание таблиц (create table).

```
CREATE TABLE public.user (  
    id bigint NOT NULL DEFAULT nextval('user_internal_id_seq'::regclass),  
    first_name character varying(512) COLLATE pg_catalog."default",  
    second_name character varying(512) COLLATE pg_catalog."default",  
    patronymic character varying(512) COLLATE pg_catalog."default",  
    email character varying(512) COLLATE pg_catalog."default",  
    phone character varying(512) COLLATE pg_catalog."default",  
    CONSTRAINT user_internal_pkey PRIMARY KEY (id)  
)  
  
CREATE TABLE public.role (  
    id bigint NOT NULL DEFAULT nextval('role_id_seq'::regclass),  
    name character varying(512) COLLATE pg_catalog."default",  
    title character varying(512) COLLATE pg_catalog."default",  
    CONSTRAINT role_pkey PRIMARY KEY (id)  
)  
  
CREATE TABLE public.user_role (  
    id bigint NOT NULL DEFAULT nextval('user_role_id_seq'::regclass),  
    user_id bigint,  
    role_id bigint,  
    CONSTRAINT user_role_pkey PRIMARY KEY (id)  
)  
  
CREATE TABLE public.task (  
    id bigint NOT NULL DEFAULT nextval('task_id_seq'::regclass),  
    title character varying(512) COLLATE pg_catalog."default",  
    description character varying(5000) COLLATE pg_catalog."default",  
    status character varying(64) COLLATE pg_catalog."default",  
    assignee_id bigint,  
    CONSTRAINT task_pkey PRIMARY KEY (id)  
)  
  
CREATE TABLE public.cart (  
    id bigint NOT NULL DEFAULT nextval('cart_id_seq'::regclass),  
    CONSTRAINT cart_pkey PRIMARY KEY (id)  
)  
  
CREATE TABLE public.product (  
    id bigint NOT NULL DEFAULT nextval('saved_product_id_seq'::regclass),
```

```

title character varying(512),
description character varying(5000),
article int,
cart_id bigint,
category_name character varying(512),
CONSTRAINT saved_product_pkey PRIMARY KEY (id)
)

```

На рисунке 12 отражена физическая модель данных в виде таблиц и отношений между ними:

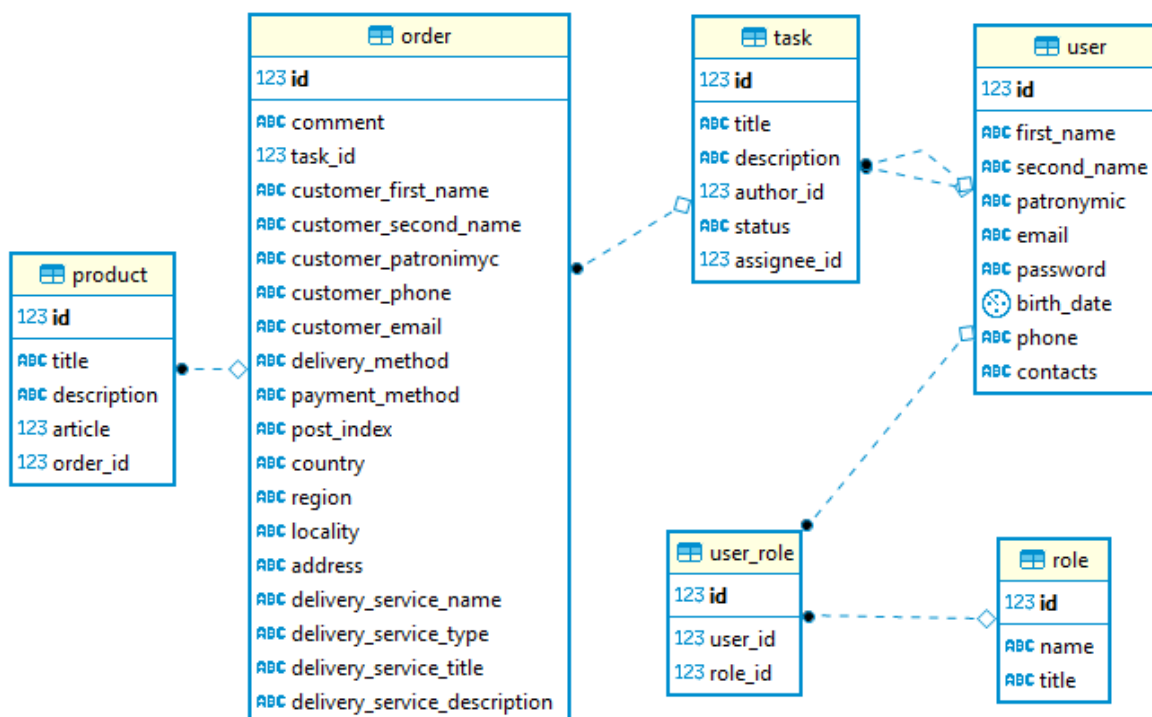


Рисунок 12 – Физическая модель данных

## 2.4 Проектирование программного обеспечения

Для наглядной демонстрации работы приложения целесообразно представить её в виде диаграмм прецедентов.

Каждый сценарий использования сосредотачивается на описании того, как достигнуть цели или задачи. Для большинства программных проектов это означает, что потребуется множество сценариев использования, чтобы определить необходимый набор свойств новой системы. Степень формальности программного проекта и его стадии будет влиять на

необходимый уровень детализации, для каждого сценария использования. Диаграмма вариантов использования, описывает функциональное назначение системы или, другими словами, то, что система будет делать в процессе своего функционирования. Диаграмма вариантов использования является исходным концептуальным представлением или концептуальной моделью системы в процессе ее проектирования и разработки [15].

На рисунке 13 представлена диаграмма поведения пользователя при работе пользователя при работе личным профилем и списком пользователей.



Рисунок 13 – Диаграмма поведения пользователя при работе с пользователями и личным кабинетом.

На диаграмме поведения (рисунок 13) в роли пользователя может быть любой авторизованный сотрудник компании, а в случае действия «Авторизация в системе» неавторизованный.

В таблице 7 описан прецедент входа в систему. Данное действие необходимо для разграничения ролей в системе среди сотрудников компании, а также для обеспечения конфиденциальности при выполнении действий каждым сотрудником.

Таблица 7 – Прецедент использования «Вход в систему».

№ п/п	Прецедент использования	
1	Название сценария	Вход в систему
2	Пользователь	Любой сотрудник организации
3	Предусловия	Пользователь неавторизован в системе
4	Основной поток	<ol style="list-style-type: none"> <li>1. Пользователь переходит на любую страницу веб-приложения</li> <li>2. Система отображает поле для ввода адреса электронной почты и пароля, а также кнопку для входа</li> <li>3. Пользователь вводит авторизационные данные в обязательные поля на странице и нажимает на кнопку «Вход»</li> <li>4. Система перезагружает страницу и отображает главную страницу веб-приложения</li> </ol>

В таблице 8 описан прецедент просмотра профиля пользователя. Указанная в таблице совокупность действий необходима как для просмотра личной информации, с целью её переноса на другие носители информации, так и для просмотра контактной информации о других сотрудниках в той же организации.

Таблица 8 – Прецедент использования «Просмотр профиля пользователя».

№ п/п	Прецедент использования	
1	Название сценария	Просмотр профиля пользователя
2	Пользователь	Любой сотрудник организации
3	Предусловия	Пользователь авторизован в системе

Продолжение таблицы 8

4	Основной поток	<ol style="list-style-type: none"> <li>1. Пользователь переходит по ссылке «Пользователи»</li> <li>2. Система отображает страницу со списком пользователей</li> <li>3. Пользователь находит в списке свою запись по фамилии имени и отчеству или запись другим интересующим именем и переходит по ссылке расположенной в записи</li> <li>4. Система отображает страницу с данными пользователя</li> </ol>
---	----------------	---

На рисунке 14 представлена диаграмма поведения системного администратора, руководителя и специалиста по трёхмерной печати при работе с задачами.

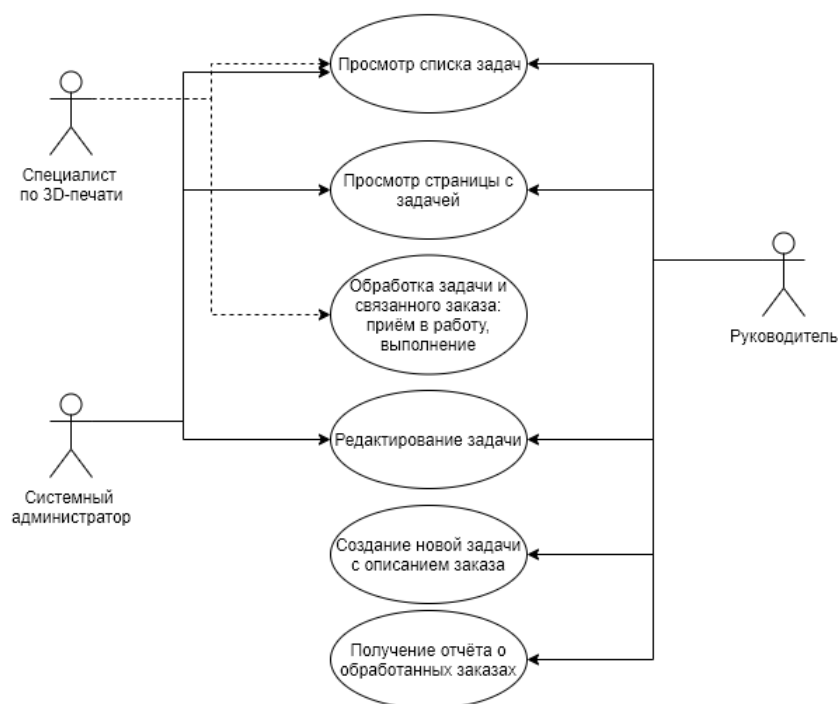


Рисунок 14 – Диаграмма поведения системного администратора и исполнителя при работе с задачами.

В таблице 9 описан прецедент просмотра задачи. Последовательность действий в данном прецеденте требуется для ознакомления с содержанием задачи и связанным с ней заказом.

Таблица 9 – Прецедент использования «Просмотр задачи».

№ п/п	Прецедент использования	
1	Название сценария	Просмотр задачи
2	Пользователь	Любой сотрудник организации
3	Предусловия	Пользователь авторизован в системе
4	Основной поток	<ol style="list-style-type: none"> <li>1. Пользователь переходит по ссылке меню «Задачи»</li> <li>2. Система отображает страницу со списком задач</li> <li>3. Пользователь находит в списке задач интересующую по наименованию переходит по ссылке расположенной в записи</li> <li>4. Система отображает страницу с детальной информацией о задаче и заказе</li> </ol>

В таблице 10 описан прецедент обработки задачи и связанного заказа. Последовательность действий в данном прецеденте требуется для продвижения задачи по жизненному циклу.

Таблица 10 – Прецедент использования «Обработка задачи и связанного заказа»

№ п/п	Прецедент использования	
1	Название сценария	Обработка задачи и связанного заказа
2	Пользователь	Специалист по 3D-печати
3	Предусловия	Пользователь авторизован в системе Пользователь находится на странице просмотра задачи

Продолжение таблицы 10

4	Основной поток	<p>1. В зависимости от текущего статуса задачи пользователь нажимает на кнопку: «В работу», «Выполнить», «Заккрыть»</p> <p>2. Система обновляет страницу и отображает страницу задачи с новым статусом</p>
---	----------------	--

В таблице 11 описан прецедент редактирования задачи.

Последовательность действий в данном прецеденте требуется для изменения данных задачи

Таблица 11 – Прецедент использования «Редактирование задачи»

№ п/п	Прецедент использования	
1	Название сценария	Редактирование задачи
2	Пользователь	<p>– Руководитель</p> <p>– Администратор</p>
3	Предусловия	<p>Пользователь авторизован в системе</p> <p>Пользователь находится на странице просмотра задачи</p>
4	Основной поток	<p>1. Пользователь нажимает на кнопку «Редактировать»</p> <p>2. Система переводит страницу задачи в режим редактирования, где позволяет изменять значения полей: «Заголовок», «Описание», «Назначена»</p> <p>3. Пользователь изменяет данные в полях и нажимает на кнопку «Сохранить»</p> <p>4. Система обновляет страницу и переводит её в режим просмотра</p>

В таблице 12 описан прецедент создания задачи. Последовательность действий в данном прецеденте требуется для создания новой задачи с данными о заказе, которые поступили в ходе обработке электронного письма или звонка на рабочий телефон компании.

Таблица 12 – Прецедент использования «Создание задачи»

№ п/п	Прецедент использования	
1	Название сценария	Создание задачи
2	Пользователь	– Руководитель
3	Предусловия	Пользователь авторизован в системе Пользователь находится на странице списка задач
4	Основной поток	<ol style="list-style-type: none"> <li>1. Пользователь нажимает на кнопку «Создать»</li> <li>2. Система открывает страницу создания задачи, где предлагает ввести обязательные поля.</li> <li>3. Пользователь вводит данные задачи в полях и нажимает на кнопку «Сохранить»</li> <li>4. Система обновляет страницу и переводит её в режим просмотра</li> </ol>

### 2.4.1 Структура программной системы

Данное веб-приложение имеет в основе трехзвенную архитектуру. Данная архитектурная модель программного комплекса предполагает наличие в нём трёх компонентов: клиента, сервера приложений (к которому подключено клиентское приложение) и сервера баз данных (с которым работает сервер приложений) [10].

Браузер, установленный на рабочем компьютере сотрудника, выполняет роль клиента, сервером служит удалённый веб-сервер с



установленным в нем веб-приложением. В качестве системы хранения данных используется СУБД PostgreSQL.

Взаимодействие звеньев архитектуры происходит следующим образом:

- Пользователь запускает браузер и вводит адрес веб-приложения;
- Браузер взаимодействует с веб-сервером посредством протокола HTTP, отправляя и принимая текстовую информацию;
- На стороне сервера происходит обработка входящих запросов от клиента и выполнение бизнес-логики. Также на стороне сервера происходит обращение к СУБД посредством подключения через JDBC и вызовом SQL запросов для изменения или получения данных;
- После обработки запроса сервер возвращает на сторону клиента структурированные текстовые данные.

На рисунке 15 отображена схема взаимодействия звеньев трёхуровневой архитектуры.

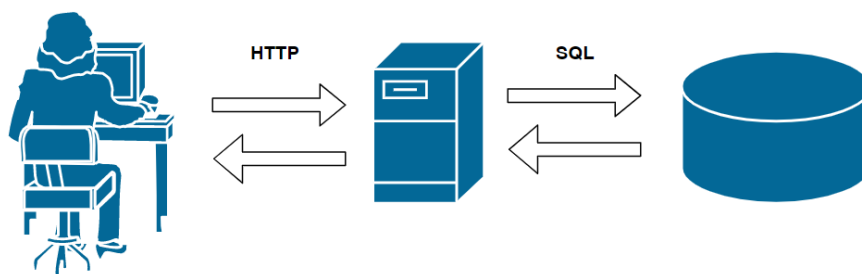


Рисунок 15 – Трёхуровневая архитектура и взаимодействие её звеньев

На рисунке 16 отражена диаграмма развертывания, которая показывает структуру программной системы, а также взаимоотношение её элементов:

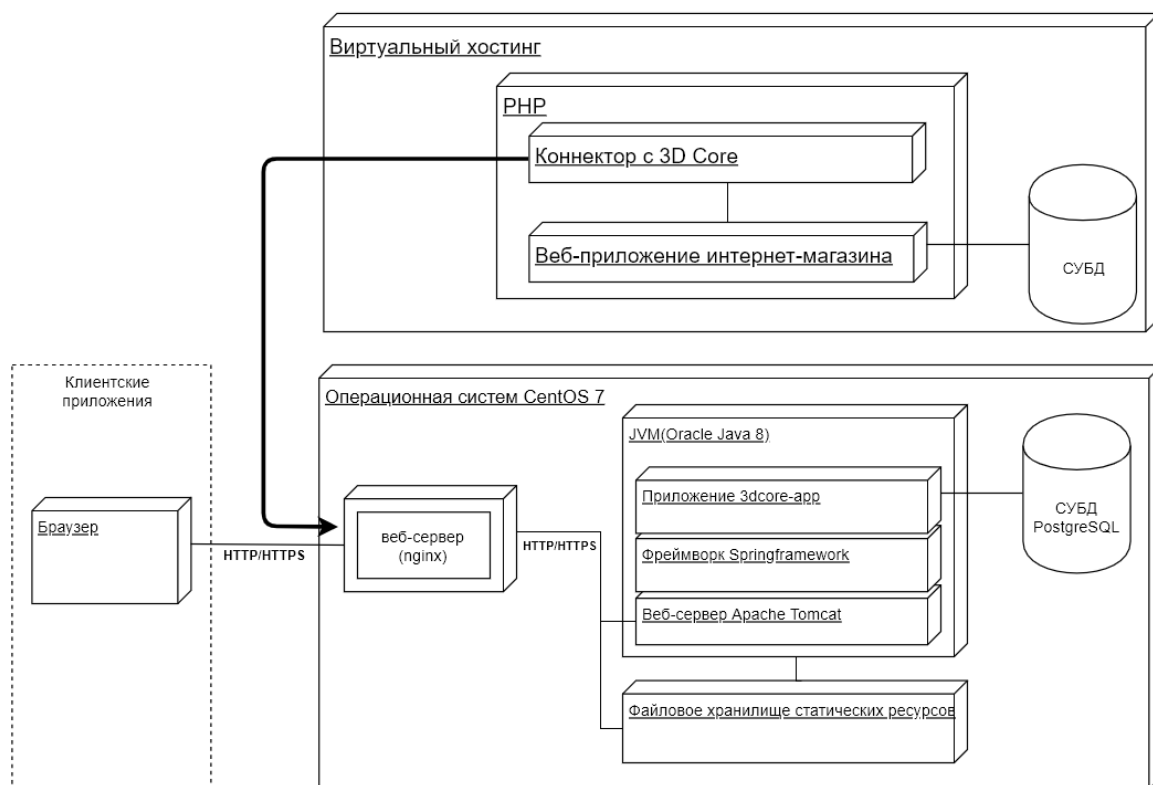


Рисунок 16 – Диаграмма развёртывания

## 2.5 Руководство программиста

Наименование системы: «Веб-приложение по автоматизации учета и обработки клиентских заказов в компании «3Д Коре».

Ключевое назначение системы:

- Обработка заказов, поступающих из интернет-магазина и созданных вручную в веб-приложении;
- Просмотр и управление задачами и связанными с ними заказами для создания изделий по списку товаров из заказа;
- Возможность формировать отчётность по принятым и выполненным заказам;
- Ведение статистики выполненных заказов.

### 2.5.1 Требования к техническому обеспечению

Требования к серверной части устанавливаются на момент запуска программного средства с учётом количества сотрудников и их активности при решении повседневных задач.

Список составляющих:

- Процессор с частотой не менее 2 ГГц (рекомендуется от 3 ГГц), количество ядер не менее 2;
- Оперативная память не менее 1 Гб (рекомендуется от 2 Гб);
- Место на жестком диске от 5 Гб.

Клиентская часть должна быть обеспечена компьютером с возможностью подключаться к сети интернет и корректно отображать веб-страницы в таких браузерах как Google Chrome, Mozilla Firefox, MS Internet Explorer, MS Edge.

### 2.5.2 Требования к программному обеспечению

Согласно выводам по проведённому анализу (п. 1.4 – 1.7), среда разработчика должна включать программное обеспечение, приведённое в таблице 13.

Таблица 13 – Среда разработки

Среда разработки	Intellij IDEA не ранее версии 2018.1
Язык программирования	Java 1.8
Операционная система	Linux Ubuntu, Linux Mint, Windows 7, 8, 10

На продуктивной среде заказчика требуется наличие программных средств (таблица 14) для корректного развёртывания и работы веб-приложения

Таблица 14 – Продуктивная среда:

Операционная система	Linux CentOS 7
Среда выполнения	Java 1.8

Продолжение таблицы 14

Java веб-сервер	Apache Tomcat 8.5
Веб-сервер для раздачи статических ресурсов	Nginx 1.16
Сервер баз данных	PostgreSQL 11

### 2.5.3 Установка системы

Процесс установки системы:

На локальном или выделенном сервере установить операционную систему CentOS 7.

Согласно инструкции «Установка и настройка CentOS 7», необходимо выполнить следующие шаги: [11]

1. Загрузка дистрибутива;
2. Создание загрузочного накопителя;
3. Запуск ПК с загрузочного накопителя;
4. Предварительная настройка;
5. Разметка дисков;
6. Завершение установки.

Согласно диаграмме развёртывания, показанной на рисунке 15, на операционную систему необходимо установить следующее программное обеспечение:

- Apache Tomcat 8.5 [12]:
  1. Обновление операционной системы CentOS;
  2. Установка Java 1.8;
  3. Создание выделенного пользователя tomcat;
  4. Загрузка дистрибутива Apache Tomcat 8.5;
  5. Установка прав доступа на ресурсы веб-сервера;

6. Настройка параметров автозапуска веб-сервера;
7. Запуск и тестирование Apache Tomcat;
8. Настройка портов для взаимодействия с веб-сервером;
- Nginx[13]:
  1. Установка Nginx при помощи менеджера пакетов;
  2. Тестирование запуска, остановки и перезагрузки веб-сервера nginx;
  3. Настройка портов для взаимодействия с веб-сервера с браузером клиента и интернет-магазином;
  4. Конфигурирование nginx.conf;
- PostgreSQL[14]:
  1. Установка PostgreSQL при помощи менеджере пакетов;
  2. Создание и инициализация базы данных **3dcore\_db** при помощи команды initdb;
  3. Создание таблиц, согласно физической модели хранения данных
  4. Настройка портов для взаимодействия базы данных только с веб-приложением на базе Apache Tomcat;
  5. Запуск и тестирование настроенной СУБД.

#### **2.5.4 Общие характеристики веб-приложения**

Система рассчитана на круглосуточный режим работы в течение 365 дней в году.

Средствами контроля и мониторинга являются автоматизированные встроенные в платформу программные компоненты, которые самостоятельно выполняют следующие действия в случае ошибки

Система, средствами операционной системы автоматически перезапускает себя в случае не критической ошибки

Посылает администратору сообщение об ошибке в письме на электронную почту

Записывают ошибку в журнал, который находится в файловой системе в файле `/var/log/apps/3dcore.log`

В случае невозможности совершить вышеуказанные действия, система выведет администратору ошибки прямо в браузер с конкретными данными о причине и расположении.

### **2.5.5 Работа с веб-приложением на стороне сервера**

Внешнее управление содержимым системы осуществляется через веб-интерфейс, описанный в документе «руководство пользователя». Для того чтобы передать полное управление всесторонними функциями портала, необходимо создать учетную запись с ролью «администратор» и по зашифрованному протоколу внести запись в СУБД, где будет содержаться логин и пароль.

Внутреннее управление порталом происходит на сервере.

Структура проекта находится в папке `/var/www/3dcore_ru` и является полноценным набором файлов и программных компонентов для управления логикой, функциональной частью системы и ее работой.

Запуск приложения происходит автоматически при старте операционной системы

### **2.5.6 Внесение изменений в веб-приложение**

Внесение изменений в работающую версию будет принято после перезапуска сервера. Перезапуск сервера может быть осуществлен как с остановкой HTTP-сервера и прерыванием соединений, так и без них.

Полная перезагрузка HTTP-сервера nginx в среде Linux выполняется с помощью команды `/etc/init.d/nginx restart`. Для мягкой перегрузки программной среды, необходимо выполнить команду `touch ~/tmp/restart.txt`.

Полная перезагрузка требуется в крайне редких случаях и в условиях эксплуатации к использованию рекомендована мягкая перегрузка.

## 2.6 Руководство пользователя

Запуск пользователем программной системы осуществляется посредством перехода в веб-браузере по корпоративному адресу my3dcore.ru

После ввода адреса в обозревателе появляется страница авторизации, см. рисунок 17, где пользователю предлагается ввести данные для авторизации: адрес электронной почты и пароль

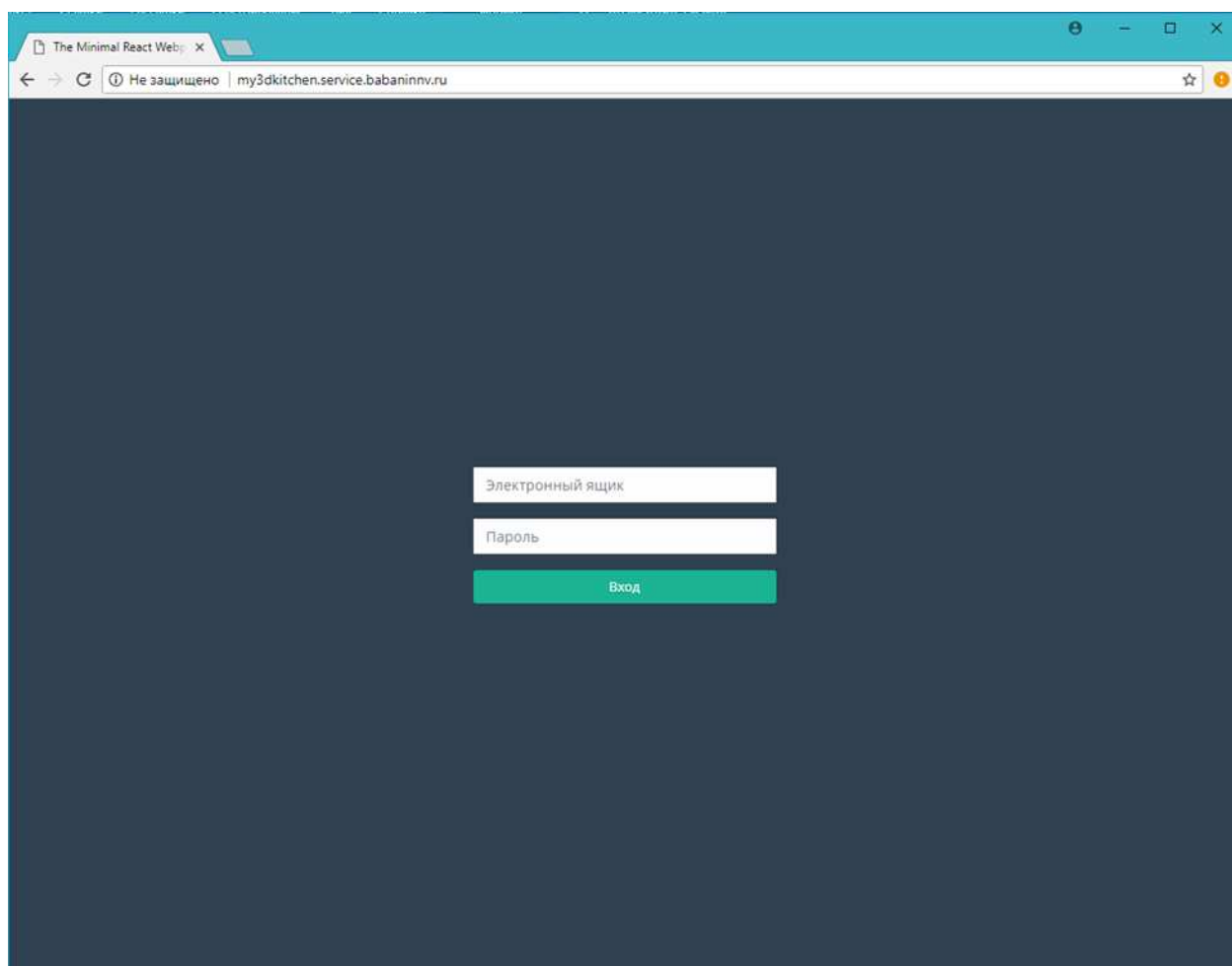


Рисунок 17 – Веб-страница для ввода адреса электронной почты и пароля

После успешного входа пользователь попадает на главную страницу, показанную на рисунке 18, где отображен список задач. Руководитель организации видит список всех задач. Если вход был осуществлён под

пользователем специалиста по трёхмерной печати, список будет содержать задачи, назначенные только на этого пользователя.

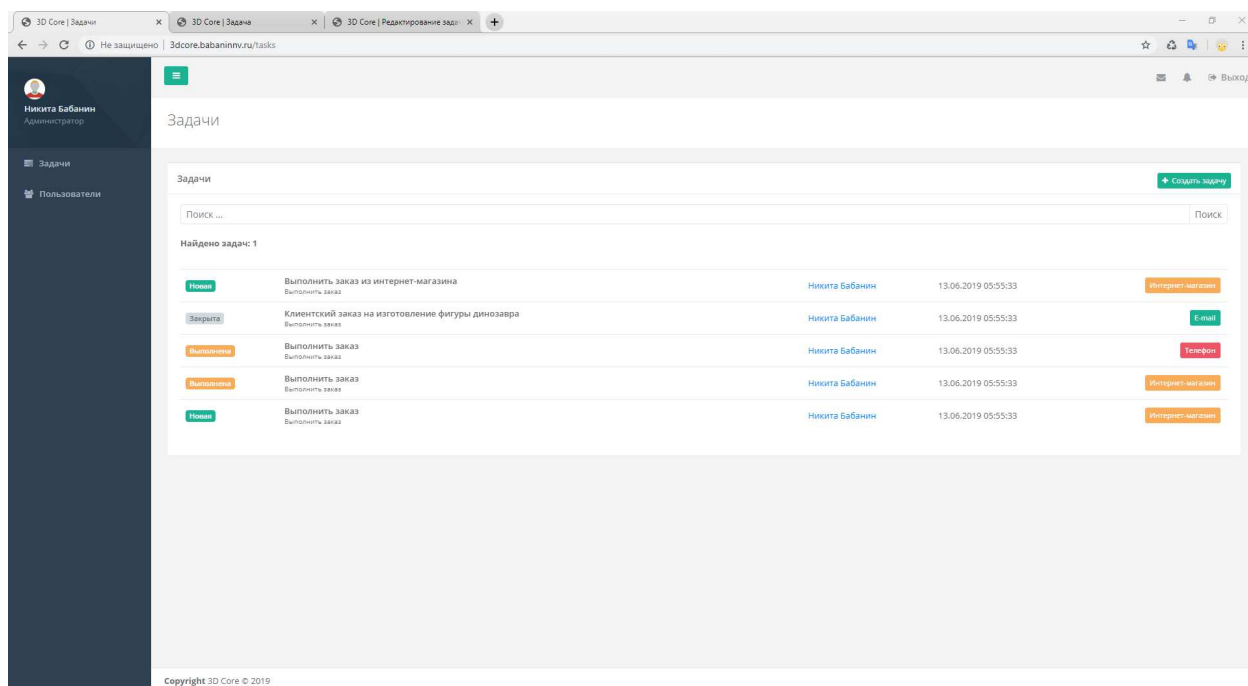


Рисунок 18 – Список задач

Чтобы попасть на страницу отображения информации о задаче необходимо перейти по ссылке кликнув на заголовок из списка задач.

Пример страницы показан на рисунке 19.

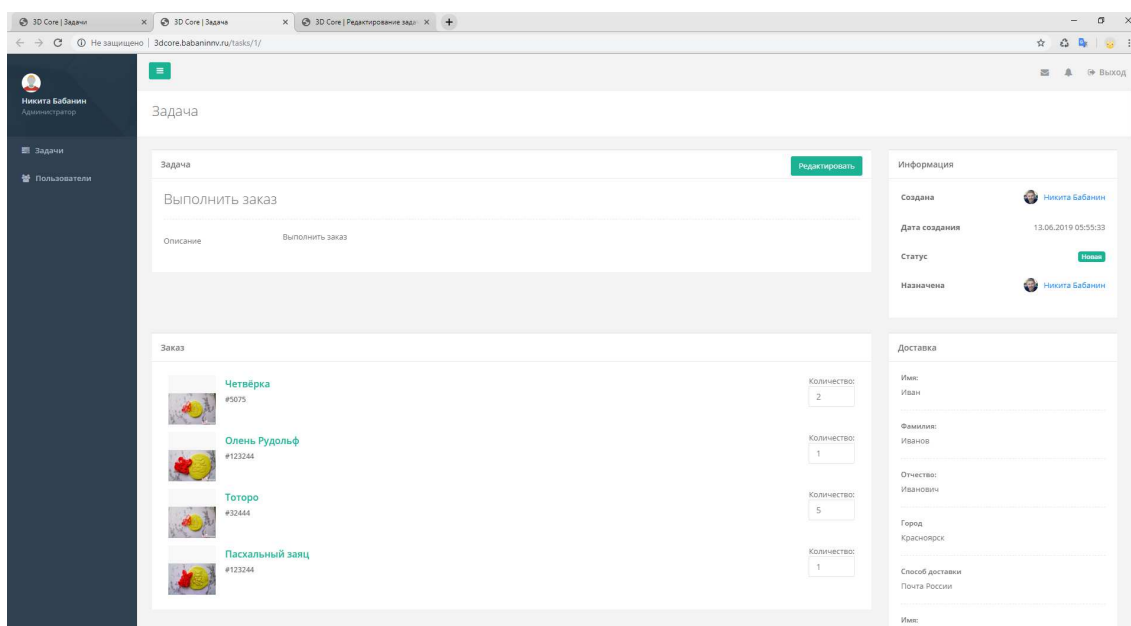


Рисунок 19 – Страница задачи



Из страницы с детальной информацией о задаче можно перейти в режим редактирования, нажав на кнопку «Редактировать», расположенную в верхнем правом углу блока «Задача»

Пример режима редактирования задачи приведён на рисунке 20

3D Core | Задачи

3D Core | Редактирование задачи

Не защищено | 3dcore.babaninmv.ru/tasks/1/edit

Никита Бабанин  
Администратор

Задачи

Пользователи

Редактирование задачи

Сохранить

Задача

Заголовок: Выполнить заказ

Описание: Выполнить работы согласно товарному списку

Информация

Создана: Никита Бабанин

Дата создания: 13.06.2019 05:55:33

Статус: Новая

Назначена: Никита Бабанин

Заказ

	Четвёрка #5075	Количество: 2
	Олень Рудольф #123244	Количество: 1
	Тоторо #32444	Количество: 5
	Пасхальный заяц #123244	Количество: 1

Доставка

Имя: Иван

Фамилия: Иванов

Отчество: Иванович

Город: Красноярск

Способ доставки: Почта России

Рисунок 20 – Страница редактирования задачи

Чтобы посмотреть список пользователей системы, необходимо перейти по ссылке «Пользователи» навигационного меню. На странице будет отображена таблица пользователей, состоящая из четырёх колонок: фотография пользователя, если таковая была загружена, имя пользователя, адрес электронной почты и текущий статус.

Пример страницы показан на рисунке 21

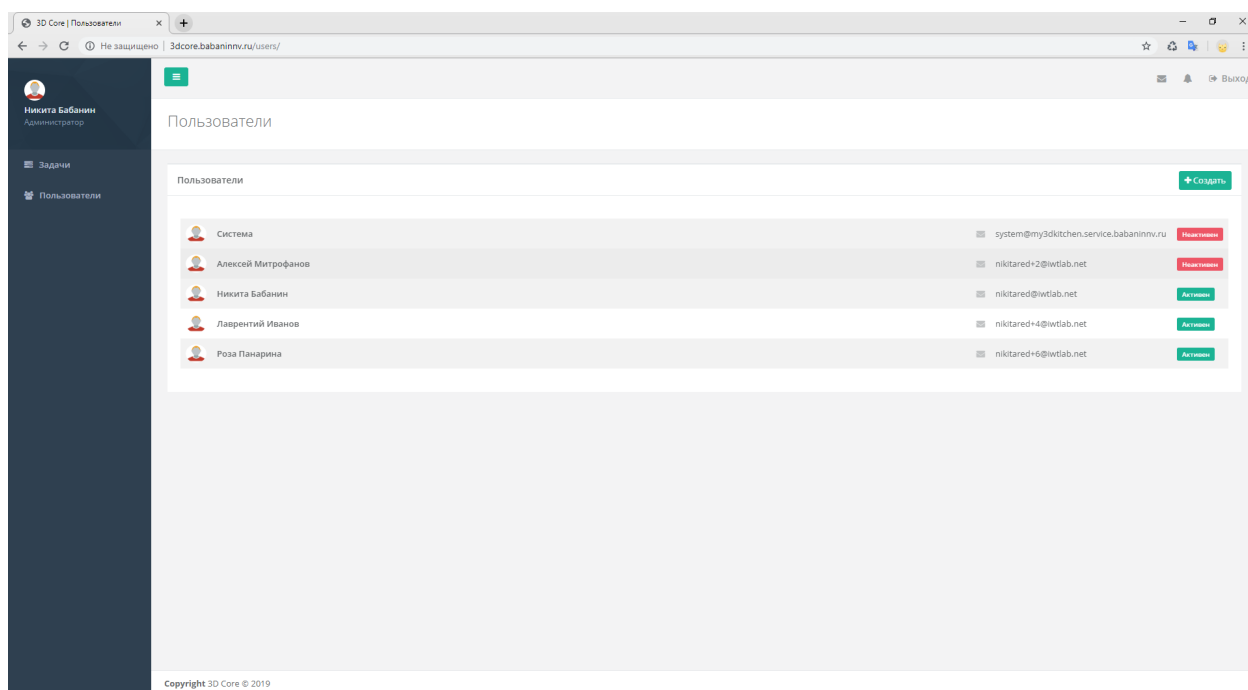


Рисунок 21 – Список пользователей

Для того чтобы посмотреть детальную информацию о пользователе необходимо перейти по ссылке, кликнув по имени пользователя.

На странице профиля в блоке «Обо мне» располагается детальная информация о пользователе: Фамилия, Имя, Отчество, E-mail, телефон

Пример страницы профиля пользователя показан на рисунке 22

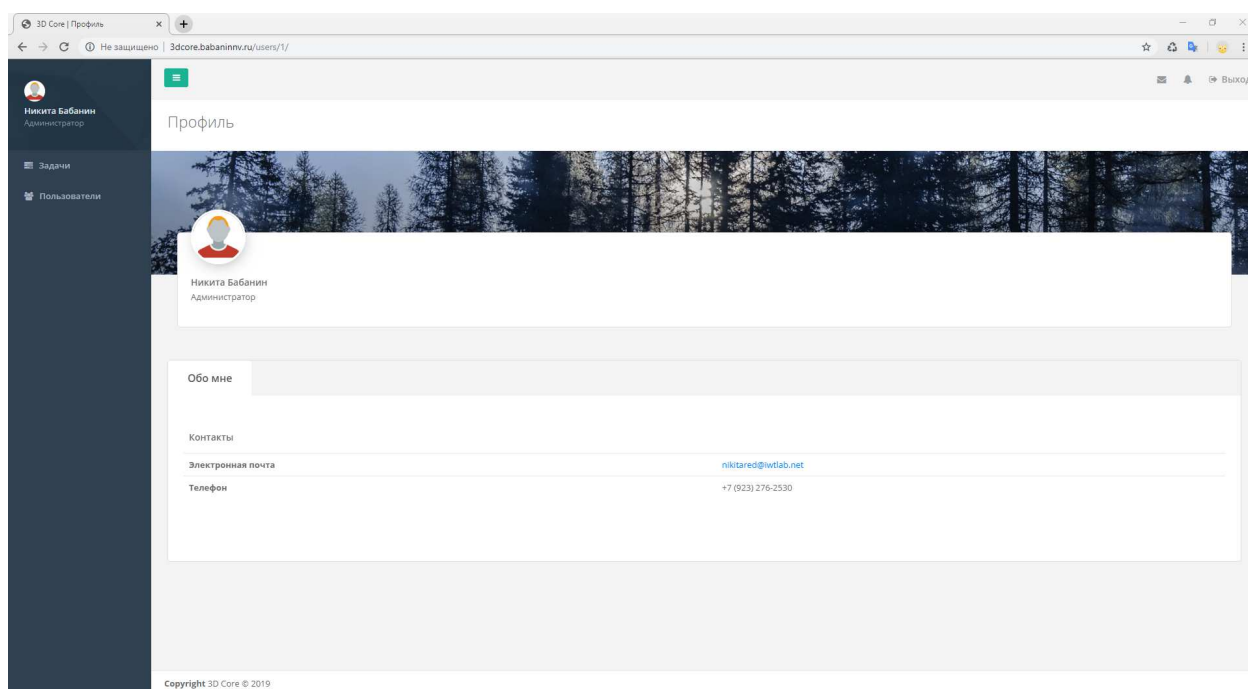


Рисунок 22 – Страница профиля пользователя

Для создания пользователя необходимо нажать на кнопку «Создать» на странице со списком пользователей. После перехода на страницу, пользователь может ввести данные и после этого нажать на кнопку «Сохранить»

Пример страницы создания показан на рисунке 23

The screenshot shows a web browser window with the URL `3dcore.babaninmv.ru/users/new/`. The page title is 'Новый пользователь' (New User). On the left, there is a dark sidebar with a user profile for 'Никита Бабанин' (Nikita Babanin) and a menu with 'Задачи' (Tasks) and 'Пользователи' (Users). The main content area has a header 'Новый пользователь' and a form titled 'Общая информация' (General Information). The form contains the following fields: 'Фамилия' (Surname) with a placeholder 'Фамилия', 'Имя' (Name) with a placeholder 'Имя', 'Отчество' (Patronymic) with a placeholder 'Отчество', 'Электронная почта' (Email) with a placeholder 'Электронная почта', and 'Роль' (Role) with a dropdown menu showing 'Выберите ...'. A green 'Сохранить' (Save) button is located at the bottom of the form. The footer of the page reads 'Copyright 3D Core © 2019'.

Рисунок 23 – Страница создания пользователя

Для того чтобы выйти из системы, необходимо нажать на ссылку «Выход». После чего страница будет перезагружена и пользователь повторно, при необходимости, может ввести свои учётные данные для авторизации

## 2.7 Тестирование программного средства

### Функциональное тестирование

Проверка разработанного прикладного программного обеспечения заключается в его функциональном тестировании с целью выявления ошибок и некорректной работы. В таблице 15 приведены результаты

функционального тестирования. В результате тестирования ошибок обнаружено не было, что позволяет сделать вывод о работоспособности разработанного прикладного программного обеспечения.

Таблица 15 – Тесты, соответствующие варианту использования.

№	Вариант использования	Тест	Результат
1	Вход в систему	Вход в систему Адрес электронной почты: admin Пароль: admin	Тест выполнен
2	Создать задачу на входящий заказ из интернет-магазина	Отправка HTTP-запроса с данными о заказе в систему 3D Core	Тест выполнен
3	Открытие и просмотр данных о задаче и связанном заказе	1. Войти в систему под администратором 2. Перейти по ссылке в одну из задач, поступивших из интернет-магазина	Тест выполнен
4	Редактирование задачи	1. Войти в систему под руководителем 2. Перейти на страницу со списком задач 3. Перейти в режим создания задачи 4. Ввести тестовые данные 5. Нажать на кнопку сохранить 6. Проверить, что страница обновилась и задача доступна для просмотра	Тест выполнен

## Модульное тестирование

Модульное или unit-тестирование позволяет провести более глубокое исследование и анализ работоспособности отдельных модулей программного обеспечения. Существуют каркасы автономного тестирования, избавляющие пользователя от ручного воспроизведения тестов. В интегрированную среду разработки, которая была использована в ходе разработки данного программного обеспечения, входит каркас автономного тестирования. Написание модульных тестов же в отличие от функционального тестирования более похоже на тестирование методом белого ящика, когда детали реализации модулей нам известны. Рассмотрим модульное тестирование на примере шести классов, отвечающих за операции с пользователями, заказами и задачами, а именно: добавление, редактирование, удаление, заказа, задачи и пользователя. Для выполнения автономного тестирования был создан и добавлен в решение тестовый проект, содержащий шесть классов. Результаты автономного тестирования представлены на рисунке 24.

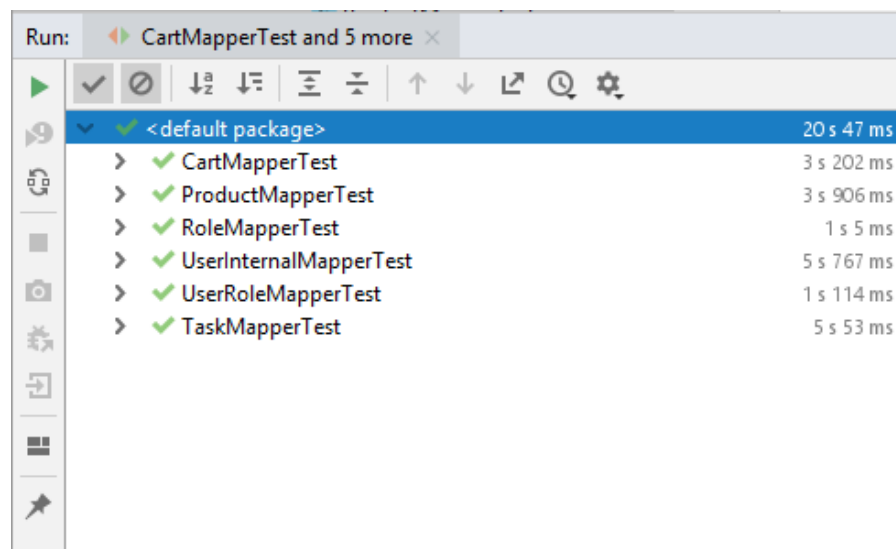


Рисунок 24 – Результаты модульных тестов

На данном рисунке в верхней части – перечень тестовых методов, статус выполнения и время их выполнения. Все тесты выполнены успешно. Общее тестирование длилось около 21 секунды. Встроенные средства IntelliJ IDEA позволяют оценить процент покрытия блоков исходного кода тестами

и сгенерировать соответствующий отчет. Анализ покрытия исходного кода представлен на рисунке 25.

Coverage: CartMapperTest and 5 more ×

↑ No coverage in package 'ru.babaninnv'

Element	Class, %	Method, %
ru.babaninnv.service	100% (0/0)	100% (0/0)
ru.babaninnv.service.my3dkitchen	0% (0/1)	0% (0/1)
ru.babaninnv.service.my3dkitchen.batis	0% (0/1)	0% (0/4)
ru.babaninnv.service.my3dkitchen.batis.ma...	100% (0/0)	100% (0/0)
ru.babaninnv.service.my3dkitchen.batis.ma...	100% (0/0)	100% (0/0)
ru.babaninnv.service.my3dkitchen.batis.result	100% (0/0)	100% (0/0)
ru.babaninnv.service.my3dkitchen.batis.resu...	95% (40/42)	90% (99/110)
ru.babaninnv.service.my3dkitchen.batis.resu...	100% (1/1)	66% (2/3)
ru.babaninnv.service.my3dkitchen.batis.resu...	100% (1/1)	100% (2/2)
ru.babaninnv.service.my3dkitchen.batis.resu...	100% (1/1)	100% (1/1)

Рисунок 25. Результаты покрытия кода

На рисунке 25 представлены результаты, как в процентном соотношении, так и в блочном. Исходя из результатов, представленных на рисунке 254, выше можно сделать вывод о том, что покрытие рассматриваемых классов находится на достаточно высоком уровне, что свидетельствует о надежности исходного кода.

## **Заключение**

Цель данной выпускной квалификационной работы состояла в автоматизации бизнес-процессов учёта и обработки клиентских заказов в компании «3Д Коре» посредством разработки и последующего внедрения веб-приложения.

При выполнении работы были получены следующие результаты.

Произведен анализ предметной области, в ходе которого рассмотрен бизнес-процесс обработки заказов из интернет-магазина путём создания электронных сообщений, передаваемых между сотрудниками для назначения конкретной задачи и для проведения отчётности. Анализ показал, что указанный бизнес-процесс не автоматизирован и может быть улучшен посредством внедрения программного средства, оптимизирующего основные процессы приёма, обработки и учёта заказов, поступающих как со стороннего ПО, которым является интернет-магазин, так и напрямую, когда клиент посылает письмо руководителю или совершает звонок в компанию. Исследование рынка готового программного обеспечения аналогичного назначения позволило сделать вывод, что целесообразно реализовать собственную программную систему, для чего был произведен выбор средств реализации веб-приложения.

На основании аналитического этапа проектирования выполнялись работы по логическому моделированию всех аспектов будущей системы, а также ее физической реализации. Приложение предназначено для выполнения в среде операционной системы Linux CentOS 7 и разработано в IDE IntelliJ IDEA. Система имеет клиент-серверную архитектуру, а также имеет возможность интегрироваться с интернет-магазином компании для оперативного получения заказов.

Разработанное приложение обладает всеми объявленными в требованиях функциями, а именно, позволяет принимать заказы из интернет-магазина, обрабатывать их, создавать задачи на исполнение. Также, задачи

могут быть созданы, вручную, используя визуальный интерфейс веб-приложения. Созданные задачи, переданные на исполнение, могут корректироваться, изменять свой статус в зависимости от этапа выполнения заказа и его отправки клиенту.

На текущий момент система внедрена в организации «3Д Коре» <sup>TM</sup> (юридическое наименование ООО «Овертайм») и находится на этапе промышленной эксплуатации.

Заказчик программного продукта отметил высокий уровень качества программного продукта, а также удобство его использования сотрудниками компании.

Руководитель организации заявил о желании развивать программный продукт и добавить в него новые возможности. Некоторые приведены в списке:

- Возможность выгружать список товаров из интернет-магазина для обработки их маркетологом
- Возможность работать с системой маркетологу
- Добавить инструментарий для работы с социальными сетями и мессенджерами
- Интегрировать в веб-приложение систему управления задачами с канбан-доской



## Список использованных источников

1. Interactive: The Top Programming Languages 2018. [Электронный ресурс] – Режим доступа: <https://spectrum.ieee.org/static/interactive-the-top-programming-languages-2018>
2. TIOBE Index for June 2019 [Электронный ресурс] – Режим доступа: <https://www.tiobe.com/tiobe-index/>
3. Возможности платформы. [Электронный ресурс] – Режим доступа: <https://www.amocrm.ru/developers/content/platform/abilities/>
4. Документация к PostgreSQL 9.6.2. [Электронный ресурс] – Режим доступа: <http://repo.postgrespro.ru/doc/pgsql/9.6.2/ru/postgres-A4-fop.pdf>
5. Справочное руководство по MySQL 4.0 [Электронный ресурс] – Режим доступа: <http://www.mysql.ru/docs/mysql-man-4.0-ru/introduction.html#features>
6. Бурмистров А.В., Белов Ю.С. Недостатки реляционных баз данных // Электронный журнал: наука, техника и образование. 2015. № 3 (3). С. 25 – 34.
7. Панченко И. PostgreSQL: вчера, сегодня, завтра // Открытые системы. СУБД. 2015. № 3. С. 34-37.
8. Eclipse, NetBeans или IntelliJ IDEA? Выбираем IDE для Java-разработки. [Электронный ресурс] – Режим доступа: <https://javarush.ru/groups/posts/1642-eclipse-netbeans-ili-intellij-idea-vihbiraem-ide-dlja-java-razrabotki>
9. TOP 10 лучших сред разработки на Java [Электронный ресурс] – Режим доступа: <https://gitjournal.tech/10-luchshih-sred-razrabotki-na-java/>
10. Трёхуровневая архитектура — Википедия [Электронный ресурс] – Режим доступа: <https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D1%91%D1%85%D1%83%D1%80%D0%BE%D0%B2%D0%BD%D0%B5%D0%B2%D0%B>

[0%D1%8F\\_%D0%B0%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0](#)

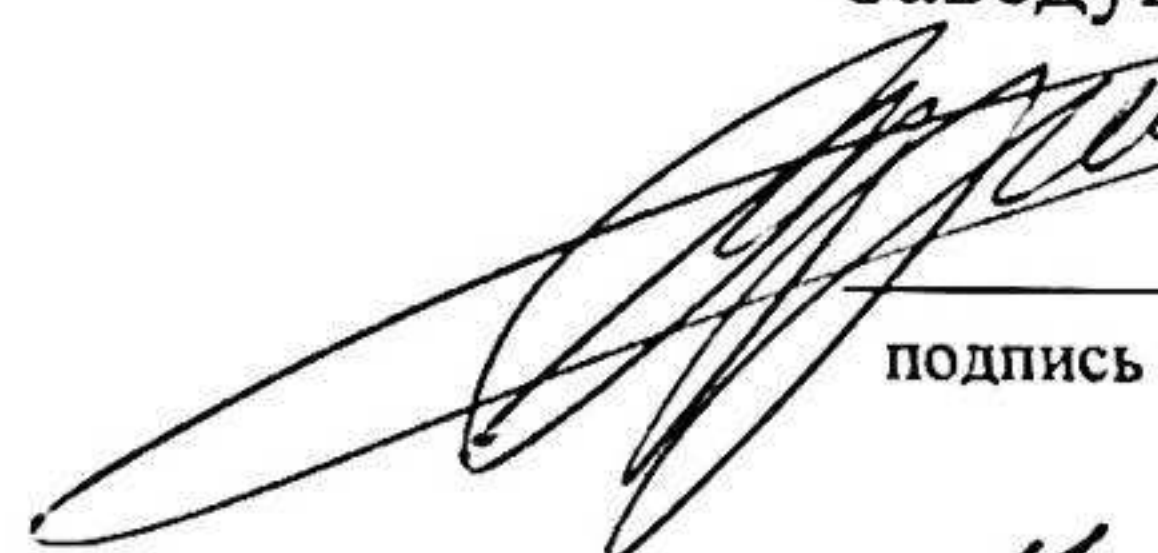
11. Установка и настройка CentOS 7 [Электронный ресурс] – Режим доступа: <http://lumpics.ru/installing-and-configuring-centos-7/>
12. How to Install Apache Tomcat 8 on CentOS 7 [Электронный ресурс] – Режим доступа: <https://www.vultr.com/docs/how-to-install-apache-tomcat-8-on-centos-7>
13. How to install and use Nginx on CentOS 7 / RHEL 7 [Электронный ресурс] – Режим доступа: <https://www.cyberciti.biz/faq/how-to-install-and-use-nginx-on-centos-7-rhel-7/>
14. How to Install PostgreSQL on CentOS 7 [Электронный ресурс] – Режим доступа: <https://www.hostinger.com/tutorials/how-to-install-postgresql-on-centos-7/>
15. Диаграмма вариантов использования (use case diagram) [Электронный ресурс] – Режим доступа: <http://khpi-iip.mipk.kharkiv.edu/library/case/leon/gl4/gl4.html>

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий  
институт

Кафедра информатики  
кафедра

УТВЕРЖДАЮ  
Заведующий кафедрой



А. С. Кузнецов

подпись

инициалы, фамилия

« 11 » 07 2019 г.

БАКАЛАВРСКАЯ РАБОТА

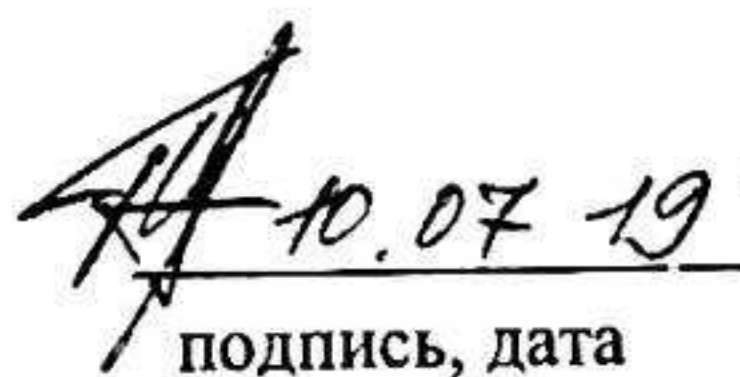
09.03.04 Программная инженерия

код - наименование направления

Автоматизация учета и обработки клиентских заказов в компании «3ДКоре»

тема

Руководитель



10.07.19

подпись, дата

доцент, канд. техн. наук

должность, ученая степень

В. В. Тынченко

инициалы, фамилия

Выпускник



10.07.19.

подпись, дата

Н. В. Бабанин

инициалы, фамилия

Нормоконтролер



10.07.19

подпись, дата

доцент, канд. техн. наук

должность, ученая степень

О. А. Антамошкин

инициалы, фамилия

Красноярск 2019



Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра «Информатика»

УТВЕРЖДАЮ

Заведующий кафедрой

А. С. Кузнецов

инициалы, фамилия

подпись

«21»

2019 г.

**ЗАДАНИЕ**  
**НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ**  
**в форме бакалаврской работы**



Студенту Бабанину Никите Валерьевичу  
фамилия, имя, отчество  
Группа ВКИ15-16БВВ Направление (специальность) 09.03.04  
подпись, дата код  
Программная инженерия  
наименование

Тема выпускной квалификационной работы:  
Автоматизация учета и обработки клиентских заказов в компании «3ДКоре»

Утверждена приказом по университету: № 6826/с от 2019-05-21

Руководитель ВКР: В.В. Тынченко, доцент кафедры  
«Информатика», канд. тех. н.  
инициалы, фамилия, должность, ученая степень

Исходные данные для ВКР: информация о бизнес-процессе учета и обработки клиентских заказов компании в «3Д Коре»

Перечень разделов ВКР:  
введение, анализ предметной области, разработка автоматизированной системы, заключение, список использованных источников

Перечень графического материала: презентационные слайды PowerPoint.

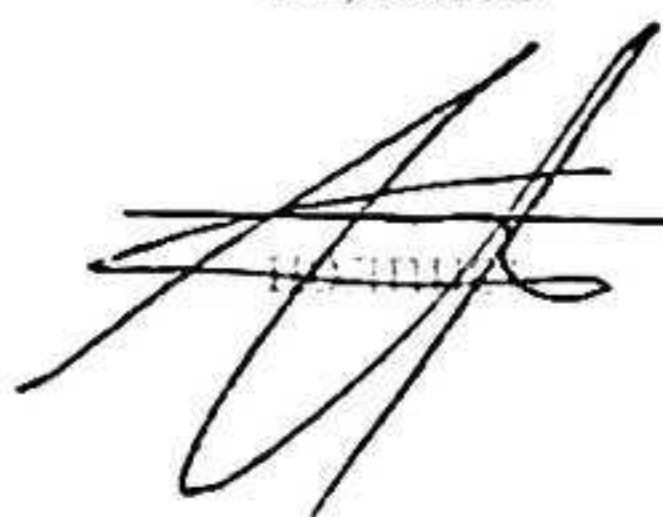
Руководитель ВКР

  
подпись

Тынченко В. В.

фамилия, инициалы

Задание принял к исполнению

  
подпись

Бабанин Н. В.

фамилия, инициалы студента

« 11 » 05 2015 г.